Università degli studi di Napoli "Federico II"

Corso di Reti di Calcolatori

Prof. Giorgio Ventre

Anno 2000/2001

Bluetooth e 802.15

Avallone Stefano

Caputo Antonio

Di Grazia Domenico

1. Introduzione

Negli ultimi decenni il progresso della microelettronica e l'utilizzo della tecnologia VLSI (Very Large Scale Integration) hanno portato ad un maggiore utilizzo di prodotti commerciali quali PC, laptop, PDA, cellulari, cordless e periferiche in genere. L'interoperabilità fra queste device fino ad oggi è stata in larga parte realizzata con collegamenti basati sull'utilizzo di cavi, con tutte le limitazioni che questo comporta (ad es. range ridotto, ingombro, estetica etc.).

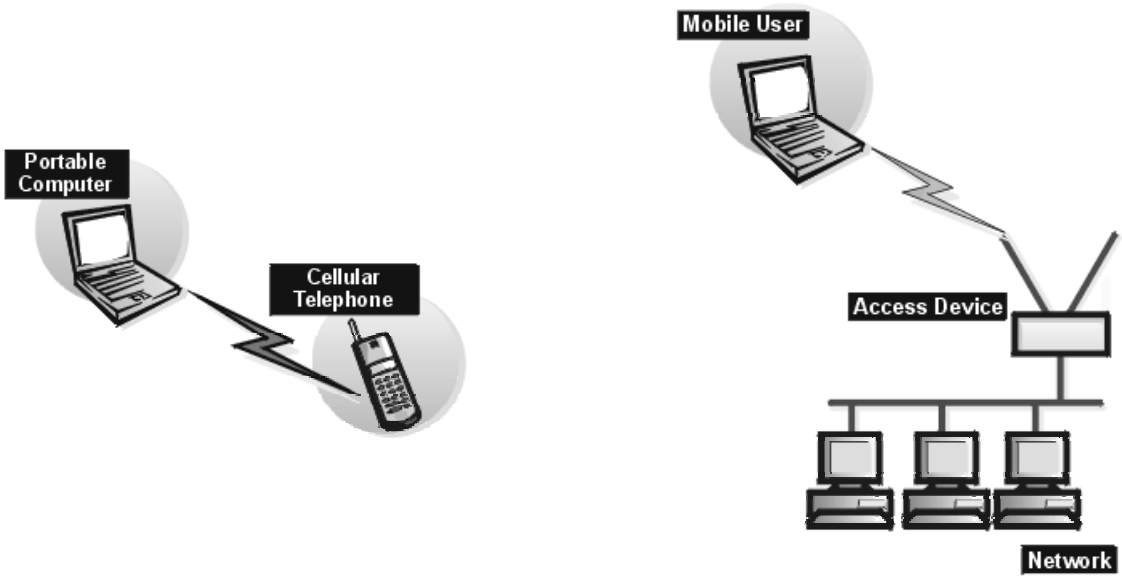
Una possibile soluzione per eliminare l'impiego dei cavi è la tecnologia IrDA che consiste nell'utilizzo di raggi infrarossi per connettere dispositivi diversi. Il maggior svantaggio di questa soluzione risiede nella necessità di disporre i dispositivi in modo tale da essere in vista.

Agli inizi degli anni '90 Ericsson cominciò un progetto di ricerca al fine di risolvere tali inconvenienti utilizzando onde radio; l'argomento si dimostrò talmente vasto da spingere la multinazionale a coinvolgere anche altri partner di rilievo (Nokia, IBM, Toshiba, Intel, Microsoft etc) in modo da poter creare uno standard de facto. Questa collaborazione ha dato vita al SIG (Special Interest Group) che ad oggi conta più di 1900 soci. Tutto questo clamore ha suscitato anche l'interesse dell' IEEE che ha deciso di standardizzare tale soluzione sotto il nome di 802.15, che si va ad aggiungere alla famiglia degli standard 802.x per le reti locali nella sua specificità di standard per reti personali.

La tecnologia Bluetooth è stata pensata per realizzare dispositivi a basso consumo, economici (al momento intorno ai 10$), poco ingombranti in modo da poter essere facilmente integrati anche in device portatili e consentire tassi di trasmissione fino a 1 Mbps (nominale).

Bluetooth è nato essenzialmente come uno stack di protocolli per la connettività wireless dei dispositivi prima citati (ed altri quali addirittura forni, lavatrici e frigoriferi) ed oggi ha l'ambizione di creare le cosiddette WPAN (Wireless Personal Area Network), ovvero microambienti nei quali i dispositivi (anche di uso comune) interagiscono tra di loro in modo del tutto trasparente all'utente.

Gli scenari applicativi sono i più disparati e vanno dalla possibilità di scambiare informazioni in modo automatico tra un PDA (Personal Digital Assistant) oppure un laptop e un desktop in modo da sincronizzare il lavoro, interconnettersi con una LAN , fino alla possibilità di utilizzare un telefonino con scheda Bluetooth come telecomando universale ad esempio per comandare elettrodomestici a distanza (10m), così come mostrato all'ultima edizione dello SMAU.

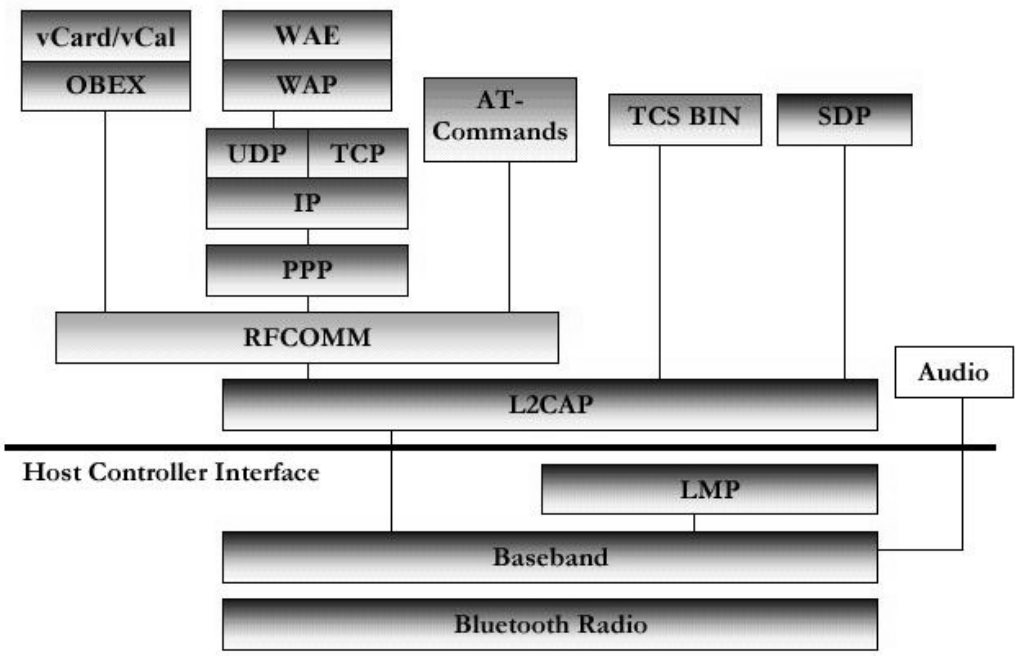


Lo standard prevede una portata di 10 m , tuttavia aumentando la potenza trasmessa da 1 mW a 100 mW si può coprire una distanza di 100 m rendendo possibili nuovi scenari applicativi quali ad esempio la possibilità di connettersi in modalità wireless ai servizi internet all'interno di stazioni ed aeroporti.

Le previsioni di penetrazione nel mercato sono eclatanti : si pensa che nei prossimi due anni verranno prodotti oltre 200 milioni di device Bluetooth compatibili. Sono inoltre in fase di studio versioni che estenderanno il tasso di trasmissione fino ai 20 Mbps ed oltre, in modo da rendere competitiva la tecnologia anche nei confronti delle wireless LAN.

Nel seguito presenteremo la pila di protocolli Bluetooth, dapprima in chiave del tutto generale per poi approfondire i dettagli delle singole parti. Infine verrà presentato un breve confronto tra lo standard Bluetooth e il futuro 802.15 .

2. Lo Stack



Lo stack Bluetooth arriva fino al livello DLC del modello OSI (strati Radio, Baseband, Lmp, L2cap) e contiene uno strato di adattamento (RFCOMM) per interfacciare le applicazioni esistenti (WAP) e quelle a venire, uno di signalling (TCS) per la gestione della telefonia, uno di servizio (SDP) che permette di conoscere i servizi offerti dai dispositivi ed infine uno strato (Audio) per la codifica del segnale vocale.

L'Host Controller Interface fornisce un'interfaccia per accedere alla parte hardware del dispositivo composta dagli strati ad esso sottostanti. Esso è dunque il mezzo attraverso il quale i livelli superiori sfruttano i servizi offerti da Baseband e Lmp.

Nella figura sono presenti anche due possibili profili applicativi che prevedono Bluetooth come mezzo per lo scambio di dati da tessere magnetiche (Bancomat, carte di credito etc.) o per interfacciare servizi WAP.

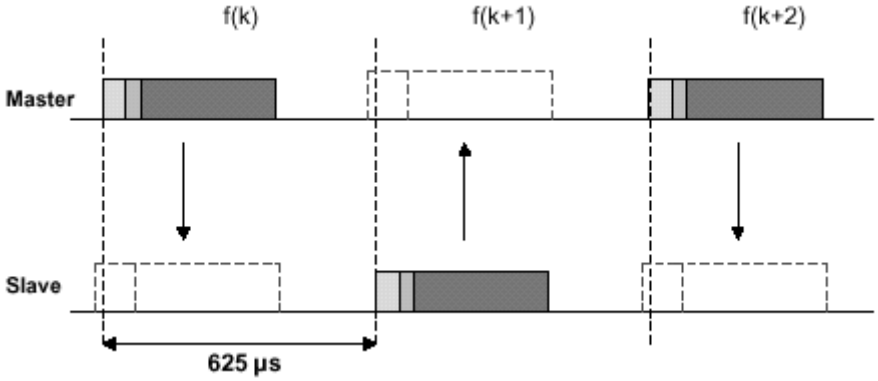
3. Radio

Lo strato Radio si occupa del trasporto fisico dei dati, quindi è l'equivalente dello strato fisico nell'architettura OSI. Attualmente le specifiche prevedono che una device trasmetta una potenza di 1mW, in questo caso il funzionamento è garantito in un raggio di 10m, tuttavia è allo studio la possibilità di aumentare la potenza fino a 100mW con un conseguente incremento del range fino a 100m. La banda di frequenze utilizzate è la cosiddetta 2.4 GHz ISM (Industrial Scientific Medicine) che si estende nell'intervallo [2400, 2483.5] MHz. Tale banda è libera quindi non è necessario alcun pagamento ad enti statali, inoltre le antenne possono essere molto piccole e questo favorisce la costruzione di apparati di dimensioni ridotte. Tuttavia proprio il fatto che tale banda è libera costituisce anche uno svantaggio, infatti anche altre tecnologie la sfruttano (si pensi ad esempio alle Wireless LAN) e quindi si possono avere dei problemi di interferenza. Si è optato allora per una tecnica di accesso multiplo detta CDMA (Code Division Multiple Access) che garantisce allo stesso tempo una buona immunità alle interferenze (anche quelle ad esempio derivanti da altri dispositivi Bluetooth che stanno comunicando) e una discreta sicurezza nei confronti di eventuali ascoltatori indesiderati.
Infatti il CDMA è una modulazione a spettro diffuso (spread spectrum), nella quale ogni segnale modulato è caratterizzato da un suo codice che è indispensabile ai fini della demodulazione (e questo quindi rende questa tecnica robusta nei confronti degli intrusi) inoltre i codici sono pensati in modo tale che segnali caratterizzati da codici diversi risultino essere ortogonali (e quindi in ricezione risulta facile isolare la comunicazione a cui siamo interessati dalle altre).
In particolare nel caso di Bluetooth è stata utilizzata una particolare tecnica CDMA detta FHSS (Frequency Hopping Spread Spectrum), nella quale viene utilizzata una modulazione GFSK (Gaussian-Frequency-Shift-Keying) che produce un segnale modulato a banda stretta. Ogni 625μs (cioè 1600 volte al secondo) la portante viene cambiata secondo una sequenza stabilita dal master (la banda disponibile è stata divisa in 79 sottobande larghe 1MHz ciascuna, di conseguenza abbiamo a disposizione 79 portanti diverse). Ne risulta in questo modo un segnale che se osservato su qualche slot è a spettro diffuso ed inoltre il codice che lo caratterizza è proprio la sequenza secondo la quale il master cambia la portante; è evidente infatti che se non si conosce tale sequenza è praticamente impossibile riuscire a demodulare il segnale perché non si saprà mai all'istante successivo quale portante verrà utilizzata.

4. Baseband

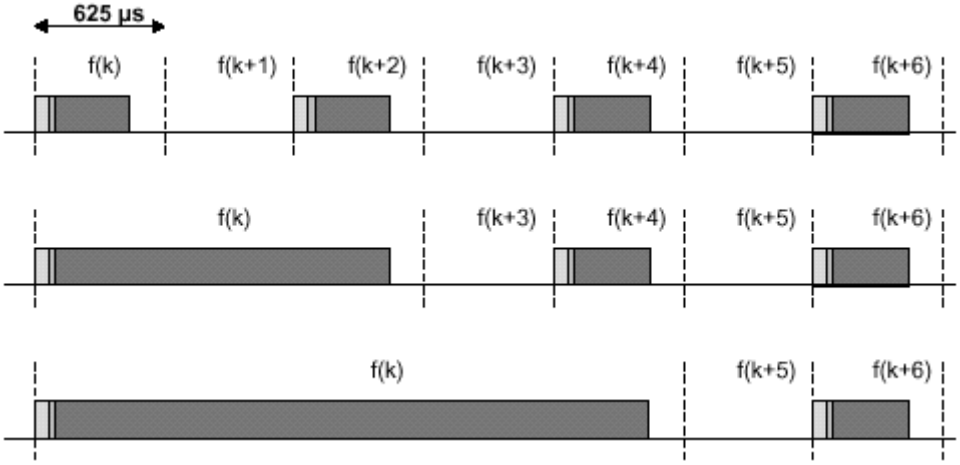
Baseband è lo strato che gestisce i canali e i collegamenti fisici offrendo una molteplicità di servizi : sincronizzazione delle unità Bluetooth, selezione della frequenza di hopping, correzione degli errori, controllo del flusso, sbiancamento dei dati e security.

Un insieme di dispositivi connessi tra di loro costituisce una piconet. Ciascuna piconet utilizza un canale suddiviso in time slot di 625µs, in ognuno dei quali è utilizzata una frequenza di hopping. La sequenza di frequenze di hopping è unica per la piconet ed è determinata dall'indirizzo del master; la fase (k), invece, è determinata dal clock del master. La sequenza di hopping deve essere tale da distribuire equamente le frequenze nella banda di 79 MHz in brevi intervalli di tempo. Lo schema di trasmissione è di tipo TDD (Time Division Duplex), in cui master e slave trasmettono alternativamente, come mostrato in figura :



Per evitare collisioni solo lo slave indirizzato nello slot master-to-slave è autorizzato a trasmettere nel seguente slot slave-to-master.

I pacchetti possono essere lunghi 1,3 o 5 slot. In caso di pacchetti multislot la frequenza resta quella del primo slot di trasmissione e la frequenza nello slot successivo è sempre determinata dal clock del master, come mostra la seguente figura:



Tra master e slave possono essere stabiliti due tipi di collegamento :

- Synchronous Connection-Oriented (SCO)
- Asynchronous Connection-Less (ACL)

SCO è un collegamento simmetrico, punto-punto tra il master e un singolo slave. Ad un tale collegamento sono riservati degli slot ad intervalli regolari, per cui può essere considerato una connessione a commutazione di circuito e tipicamente viene usato per trasportare voce. Questo è il motivo per cui i pacchetti SCO non vengono mai ritrasmessi. Il master può mantenere fino a 3 collegamenti SCO con lo stesso slave o con slave diversi. Uno slave può mantenere fino a 3 collegamenti SCO con lo stesso master oppure due se con master diversi.
Negli slot non riservati a collegamenti SCO il master può scambiare pacchetti con tutti gli slave attivi nella piconet instaurando collegamenti ACL, connessioni punto-multipunto a commutazione di pacchetto. Per la maggior parte dei pacchetti ACL è prevista la ritrasmissione per assicurare l'integrità dei dati ; è inoltre prevista la trasmissione in broadcast di pacchetti ACL.

In Bluetooth sono definiti 5 canali logici :

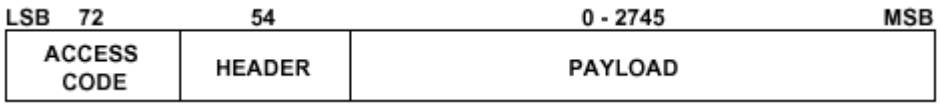
- canale di controllo LC (Link Control) : trasporta informazioni di controllo di basso livello tipo ARQ, controllo di flusso e caratterizzazione del payload; fisicamente utilizza l'header dei pacchetti
- canale di controllo LM (Link Manager) : trasporta le informazioni di controllo scambiate tra i livelli LMP del master e degli slave; è indicato dal valore 11 del campo L_CH nel payload header ed utilizza tipicamente pacchetti DM
- canali d'utente UA/UI (User Asynchronous/Isochronous data) : trasportano pacchetti L2CAP contenenti dati d'utente asincroni o isocroni; fisicamente vengono utilizzati uno o più pacchetti baseband (come indicato nel campo L_CH) in collegamenti ACL o SCO (pacchetto DV)
- canale d'utente US (User Synchronous data) : trasporta dati d'utente sincroni; utilizza collegamenti SCO

Ogni unità Bluetooth possiede un indirizzo di 48 bit (BD_ADDR, Bluetooth Device Address) derivato dagli standard IEEE 802 e diviso in tre parti : LAP (lower address part,24 bit), UAP (upper address part,8 bit), NAP (non-significant address part,16 bit) .

4.1 Il pacchetto

Passiamo ad analizzare la struttura di un pacchetto, mostrata in figura :

bi

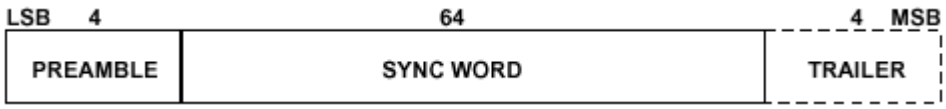


E' anche possibile avere pacchetti contenenti soltanto il campo access code (in questo caso è lungo 68 bit) oppure i campi access code e header.

Il campo access code è usato per la sincronizzazione, per la compensazione del DC offset e per l'identificazione. Sono definiti 3 tipi diversi di access code :

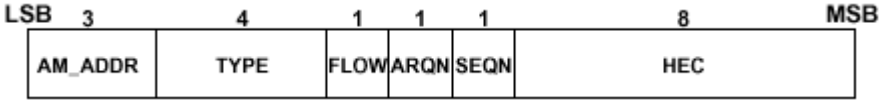
- Channel Access Code (CAC) : identifica una piconet, è usato in tutti i pacchetti scambiati sul canale di una piconet
- Device Access Code (DAC) : usato per speciali procedure di signalling, come ad esempio paging e risposta al paging
- Inquiry Access Code (IAC) : ce ne sono due tipi : general IAC (GIAC) è comune a tutti i dispositivi ed è usato per scoprire quali unità Bluetooth sono presenti; dedicated IAC (DIAC) è comune ad un gruppo di unità Bluetooth che condividono certe caratteristiche ed è usato per scoprire quali unità di questo gruppo sono presenti

La struttura del campo access code è la seguente :



Il preambolo è una sequenza zero-uno usata per facilitare la compensazione del DC offset (1010 o 0101 a seconda che il primo bit della sync word sia 1 o 0). La sync word è una parola codice di 64 bit derivata da un indirizzo di 24 bit : nel caso di CAC si usa il LAP (Lower Address Part) del master, nel caso di DAC si usa il LAP dello slave, nel caso di GIAC e DIAC si usa rispettivamente un LAP riservato e un LAP dedicato. Il modo in cui è costruita la sync word garantisce una grande distanza di Hamming tra sync word basate su LAP diversi; inoltre le buone proprietà di autocorrelazione della sync word migliorano il processo di sincronizzazione. Il trailer è una sequenza zero-uno con funzione analoga al preambolo.

L'header ha la seguente struttura :



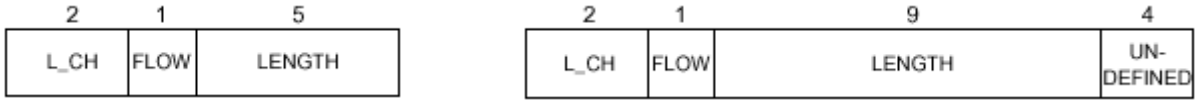
L'header consiste di 18 bit ed è utilizzata una codifica FEC 1/3 (ogni bit è ripetuto 3 volte, non mostrato in figura) pertanto la lunghezza totale è 54 bit.

AM_ADDR (Active Member Address) è un indirizzo assegnato temporaneamente ad uno slave attivo (ce ne possono essere fino a 7 contemporaneamente quindi sono sufficienti 3 bit) ed è usato in tutti i pacchetti scambiati tra il master e lo slave. L'indirizzo costituito da tutti zeri è usato nei pacchetti spediti in broadcast dal master. Gli slave rilasciano questo indirizzo quando abbandonano lo stato attivo. Il campo TYPE indica il tipo di pacchetto; ci sono 16 tipi di pacchetto e sono descritti in seguito. L'interpretazione del campo type dipende dal tipo di collegamento fisico instaurato, ACL o SCO. Il bit FLOW è usato per il controllo di flusso su collegamenti ACL. Quando il buffer in ricezione è pieno un' indicazione di stop è inviata (flow=0) per fermare la trasmissione di pacchetti. Notiamo che pacchetti di controllo (ID,POLL,NULL) e pacchetti SCO possono essere ancora ricevuti. Quando si svuota il buffer viene inviata un' indicazione di go (flow=1). Il bit ARQN è usato per informare la sorgente dell'esito della trasmissione : in caso di successo viene spedito un ACK (arqn=1) altrimenti un NAK (arqn=0), utilizzando la tecnica del piggybacking. Lo slave risponde nello slot slave-to-master immediatamente successivo allo slot in cui ha ricevuto il pacchetto mentre il master risponde nel successivo slot master-to-slave in cui spedisce un pacchetto allo stesso slave. I pacchetti spediti in broadcast dal master non prevedono un riscontro (ack) e per tale motivo vengono ritrasmessi un fissato numero di volte.

Il bit SEQN fornisce una numerazione sequenziale dei pacchetti di un flusso : per ogni nuovo pacchetto trasmesso il bit seqn viene invertito. La destinazione confronta il bit seqn del pacchetto corrente con quello del pacchetto precedente : se sono diversi vuol dire che è arrivato un nuovo pacchetto altrimenti è stato ritrasmesso il pacchetto precedente. Questo meccanismo evita la duplicazione dei pacchetti in ricezione a causa di ack persi; è quindi evidente che il bit seqn è significativo solo per i pacchetti che contengono un codice CRC, per i quali è prevista la ritrasmissione. Il campo HEC (Header Error Check) è costituito dagli 8 bit di ridondanza introdotti dalla codifica polinomiale (CRC,cyclic redundancy check) usata per controllare l'integrità dell'header. Se il controllo fallisce l'intero pacchetto viene scartato.

Il campo payload è di 2 tipi : il campo voce è presente nei pacchetti SCO, il campo dati nei pacchetti ACL (ad eccezione di un tipo di pacchetto che li ha entrambi). Il campo voce ha lunghezza fissa e non contiene il payload header. Il campo dati consiste di 3 segmenti : un payload header, un payload body ed un codice CRC.

Il payload header è lungo 1 byte in pacchetti single-slot e 2 byte in pacchetti multi-slot :



I 2 bit L_CH specificano il canale logico : il codice 11 è usato per messaggi LMP, il codice 10 è usato per il primo frammento di un pacchetto L2CAP o quando non c'è frammentazione, il codice 01 è usato per i successivi frammenti e il codice 00 è riservato per usi futuri. Il bit flow è usato per il controllo di flusso a livello L2CAP (flow=0 significa stop,flow=1 significa go); è settato e rilevato dal link manager. Il campo length indica la lunghezza in byte del payload body.

Il payload body contiene l'informazione da trasmettere e determina l'effettivo throughput.

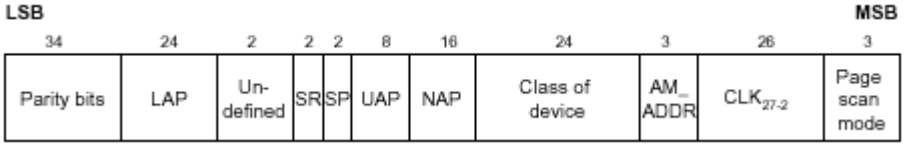
Per controllare l'informazione si usa una codifica polinomiale che aggiunge 16 bit di ridondanza.

Lo schema di ritrasmissione dei pacchetti ARQ può causare un ritardo variabile del flusso di traffico. Per alcune applicazioni è ammesso solo un limitato ritardo : le ritrasmissioni sono consentite fino ad un certo punto, oltre il quale il payload corrente deve essere scartato e il successivo payload deve essere considerato. Questo tipo di trasferimento dati è indicato come traffico isocrono; l'interruzione dello schema di ritrasmissione è realizzata mediante l'operazione di "flush" del vecchio payload. Tale operazione comporta la perdita delle parti restanti del messaggio L2CAP, per cui il successivo pacchetto, appartenente ad un nuovo messaggio L2CAP, deve avere il campo L_CH nel payload header pari a 10. Vediamo i 16 tipi di pacchetto definiti :

TYPE code $b_3b_2b_1b_0$	Slot occupancy	SCO link	ACL link
0000	1	NULL	NULL
0001	1	POLL	POLL
0010	1	FHS	FHS
0011	1	DM1	DM1
0100	1	undefined	DH1
0101	1	HV1	undefined
0110	1	HV2	undefined
0111	1	HV3	undefined
1000	1	DV	undefined
1001	1	undefined	AUX1
1010	3	undefined	DM3
1011	3	undefined	DH3
1100	3	undefined	undefined
1101	3	undefined	undefined
1110	5	undefined	DM5
1111	5	undefined	DH5

Ci sono 5 pacchetti comuni a entrambi i collegamenti fisici (ACL e SCO) :

- pacchetto ID (identity) : usato nelle procedure di paging, inquiry e relative risposte, ha una lunghezza fissa di 68 bit e consiste del solo campo access code, di tipo DAC o IAC
- pacchetto NULL : costituito dal channel access code e dal packet header, non ha payload; è usato per mandare informazioni alla sorgente riguardanti il successo di trasmissioni precedenti (arqn) o lo stato del buffer di ricezione (flow)
- pacchetto POLL : non ha payload, è usato dal master per interrogare e verificare la connessione con gli slave, che sono obbligati a rispondere anche se non hanno dati da trasmettere
- pacchetto FHS : speciale pacchetto di controllo usato nelle procedure di page e inquiry che contiene, tra le altre cose, l'indirizzo e il clock del mittente ovvero le informazioni necessarie per la sincronizzazione delle unità sulla frequenza di hopping; viene effettuata una codifica FEC 2/3 del payload (contenente un codice CRC di 16 bit), ovvero si utilizza un codice di Hamming (15,10) che aggiungendo 5 bit di parità riesce a correggere gli errori singoli e a rivelare gli errori doppi in una parola codice. Osserviamo una particolarità delle ritrasmissioni di pacchetti FHS : il payload, contenendo informazioni sul clock in tempo reale, non è lo stesso ma viene aggiornato. La struttura del payload è la seguente :



Esso contiene le tre parti dell'indirizzo del mittente (LAP,UAP,NAP) e il valore del clock di sistema del mittente campionato all'inizio della trasmissione del pacchetto. Se inviato dal master durante la procedura di page, il campo AM_ADDR contiene l'indirizzo di 3 bit che il master assegna allo slave destinatario del pacchetto FHS. I campi SR (Scan Repetition) e SP (Scan Period) indicano rispettivamente l'intervallo tra due page scan windows consecutive e la durata di tali finestre. Il campo Page scan mode indica la modalità di scan adottata dal mittente (ce ne sono una obbligatoria e tre opzionali)

- pacchetto DM1 (Data-Medium rate) : trasporta messaggi di controllo o dati d'utente. In un collegamento SCO può interrompere il flusso sincrono di dati per spedire informazioni di controllo. Per il campo payload si utilizza prima una codifica CRC (16 bit di ridondanza) e poi una codifica FEC 2/3

Ci sono altri 4 pacchetti usati nei collegamenti sincroni connection-oriented (SCO). Essi non contengono un CRC nel payload e non sono mai ritrasmessi; tipicamente sono usati per trasmettere voce a 64 kbps.

- pacchetto HV1 (High-quality Voice) : trasporta 1.25ms di conversazione ovvero, ad una frequenza di campionamento di 8 kHz, 10 campioni del segnale vocale che, quantizzati ad 8 bit per campione, risultano in 10 bytes. Ogni bit è ripetuto 3 volte (codifica FEC 1/3). Un pacchetto HV1 deve essere trasmesso ogni 2 slot
- pacchetto HV2 : trasporta 2.5ms di conversazione (20 bytes, protetti da una codifica FEC 2/3) e quindi deve essere trasmesso ogni 4 slot
- pacchetto HV3 : contiene 30 bytes di informazione (non protetti da codifica FEC) quindi può trasportare 3.75ms di voce a 64 kbps e deve essere trasmesso ogni 6 slot
- pacchetto DV : trasporta sia dati che voce; il payload è diviso in un campo voce di 80 bit e in un campo dati di massimo 150 bit. Il campo voce non è protetto da codifica FEC, il campo dati contiene 16 bit CRC e poi è codificato con la tecnica FEC 2/3. Il campo voce non viene mai ritrasmesso (cioè è sempre nuovo) mentre il campo dati può essere ritrasmesso

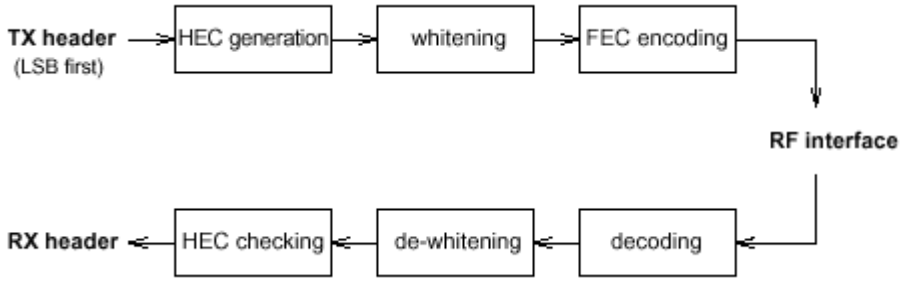
I pacchetti HV non hanno payload header, il pacchetto DV ha un payload header di 1 byte nel campo dati.

Ci sono 6 tipi di pacchetti utilizzati solo nei collegamenti asincroni connectionless (ACL) per trasportare dati d'utente o dati di controllo. In tutti i pacchetti (eccetto l'AUX1) il campo payload è codificato aggiungendo 16 bit CRC ed è prevista la ritrasmissione.

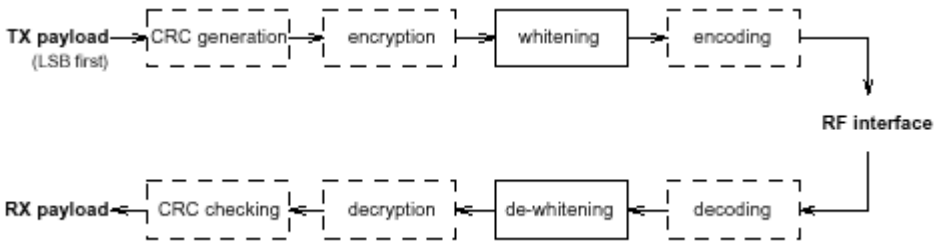
- pacchetto DH1 (Data – High rate) : simile al pacchetto DM1, con la differenza che il payload non presenta la codifica FEC. Di conseguenza contiene più bytes di informazione
- pacchetto DM3 : come il DM1 ma con un campo payload più esteso, infatti questo pacchetto può coprire fino a 3 time slot
- pacchetto DH3 : come il DM3 ma senza la codifica FEC del payload
- pacchetto DM5 : come il DM1 ma con un campo payload più esteso, infatti questo pacchetto può coprire fino a 5 time slot
- pacchetto DH5 : come il DM5 ma senza la codifica FEC del payload
- pacchetto AUX1 : come il DH1 ma senza la codifica CRC del payload

Lo scopo della codifica FEC è quello di ridurre il numero di ritrasmissioni. Tuttavia, su un canale piuttosto affidabile, dà luogo ad un overhead che riduce il throughput. Per questo motivo i pacchetti sono stata definiti in modo da consentire una certa flessibilità : per collegamenti ACL i pacchetti DM prevedono la codifica FEC, i pacchetti DH no; per collegamenti SCO i pacchetti HV1 prevedono una codifica FEC 1/3, i pacchetti HV2 una codifica FEC 2/3, i pacchetti HV3 nessuna codifica. L' header del pacchetto, invece, è sempre protetto da una codifica FEC 1/3 siccome contiene informazioni importanti.

Osserviamo che prima della trasmissione sia l'header che il payload vengono mescolati (scrambled) usando una parola sbiancante (whitening) al fine di eliminare le strutture periodiche e minimizzare l'escursione della componente continua di un pacchetto. Questa operazione viene effettuata prima della codifica FEC. In ricezione viene effettuata l'operazione inversa utilizzando la stessa parola sbiancante. I bit dell' header subiscono quindi il seguente processo :



Per questioni di sicurezza i bit del payload possono essere criptati. A tal scopo sono previste chiavi di crittografia e di autenticazione e una procedura di autenticazione. Le chiavi di autenticazione possono essere semipermanenti o temporanee : nel primo caso sono immagazzinate in una memoria non volatile, possono essere modificate e possono essere utilizzate in più sessioni (per sessione si intende l'intervallo di tempo in cui una unità è membro di una piconet); nel secondo caso possono essere utilizzate in un'unica sessione. Le chiavi di crittografia spesso derivano da quelle di autenticazione. I bit del payload subiscono quindi il seguente trattamento, dove sono tratteggiati i blocchi opzionali :



Nelle seguenti tabelle riportiamo il bit-rate relativo ai diversi pacchetti

Type	User Payload (bytes)	Symmetric Max. Rate (kb/s)
HV1	10	64.0
HV2	20	64.0
HV3	30	64.0
DV*	10+(0-9) D	64.0+57.6 D

Type	User Payload (bytes)	Symmetric Max. Rate (kb/s)	Asymmetric Max. Rate (kb/s)	
			Forward	Reverse
DM1	0-17	108.8	108.8	108.8
DH1	0-27	172.8	172.8	172.8
DM3	0-121	258.1	387.2	54.4
DH3	0-183	390.4	585.6	86.4
DM5	0-224	286.7	477.8	36.3
DH5	0-339	433.9	723.2	57.6
AUX1	0-29	185.6	185.6	185.6

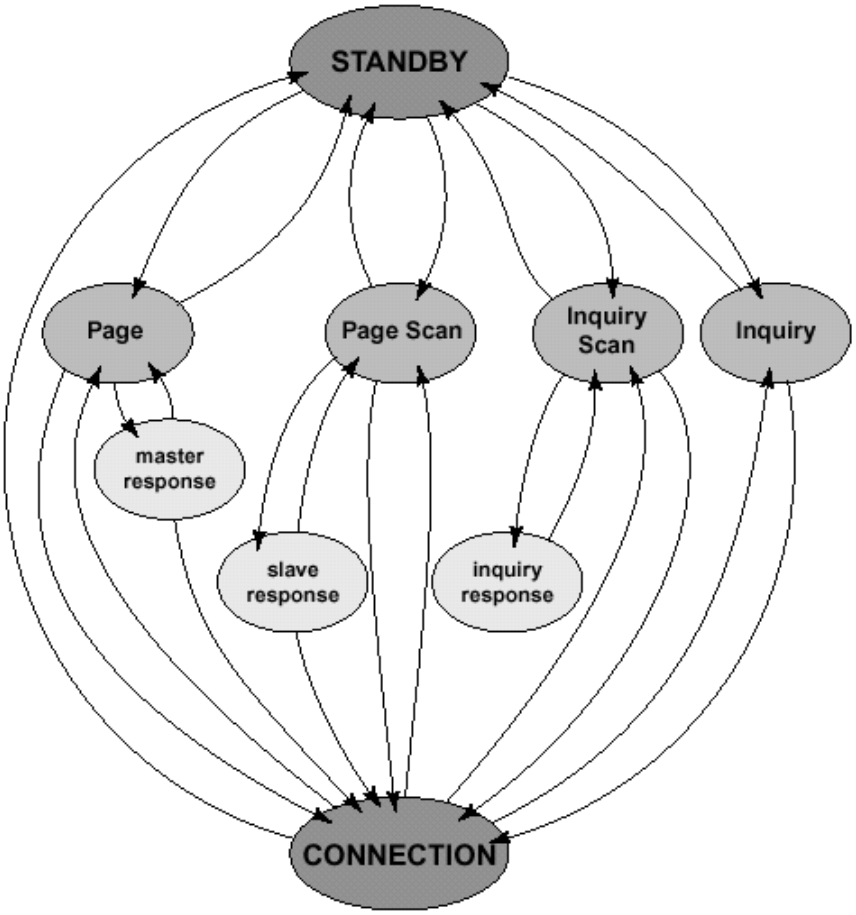
* i valori seguiti da D si riferiscono al campo dati

4.2 Piconet

Descriviamo ora come nasce una piconet e come le unità Bluetooth possono aggiungersi o abbandonare la stessa. Il canale di una piconet è completamente caratterizzato dal master : l'indirizzo BD_ADDR del master determina la sequenza di hopping e il channel access code, il suo clock determina la fase nella sequenza di hopping e la tempificazione. Per definizione, il master è l'unità Bluetooth che inizia la connessione con uno o più slave. Tutte le unità Bluetooth sono uguali, nel senso che tutte possono diventare master in una piconet.

Ogni unità Bluetooth ha un proprio clock "nativo" che non viene mai modificato; esso viene generato da un oscillatore avente un'accuratezza di ±20ppm nei periodi di attività e da un LPO (Low Power Oscillator) avente un'accuratezza di ±250ppm nei periodi di riposo. Il clock (avente una frequenza di 3.2 kHz) va in ingresso ad un contatore a 28 bit che scatta ad ogni periodo del clock (312.5µs). Quando si forma una piconet il master invia il proprio clock agli slave, i quali aggiungono un offset al proprio clock nativo al fine di sincronizzarsi col master. Tale offset va regolato periodicamente a causa di drift degli oscillatori.

Per descrivere la nascita di una piconet facciamo uso del seguente diagramma di stato :

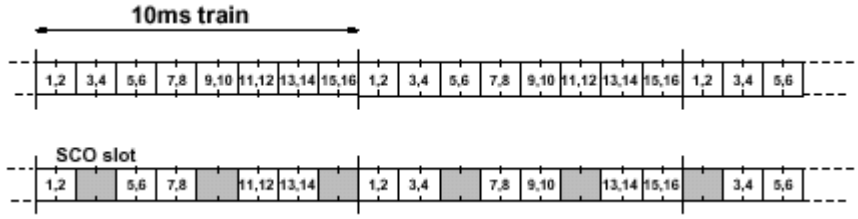


Ci sono due stati principali (*standby* e *connection*) e sette substati che servono ad aggiungere nuove unità ad una piconet.

4.2.1 Procedura di paging

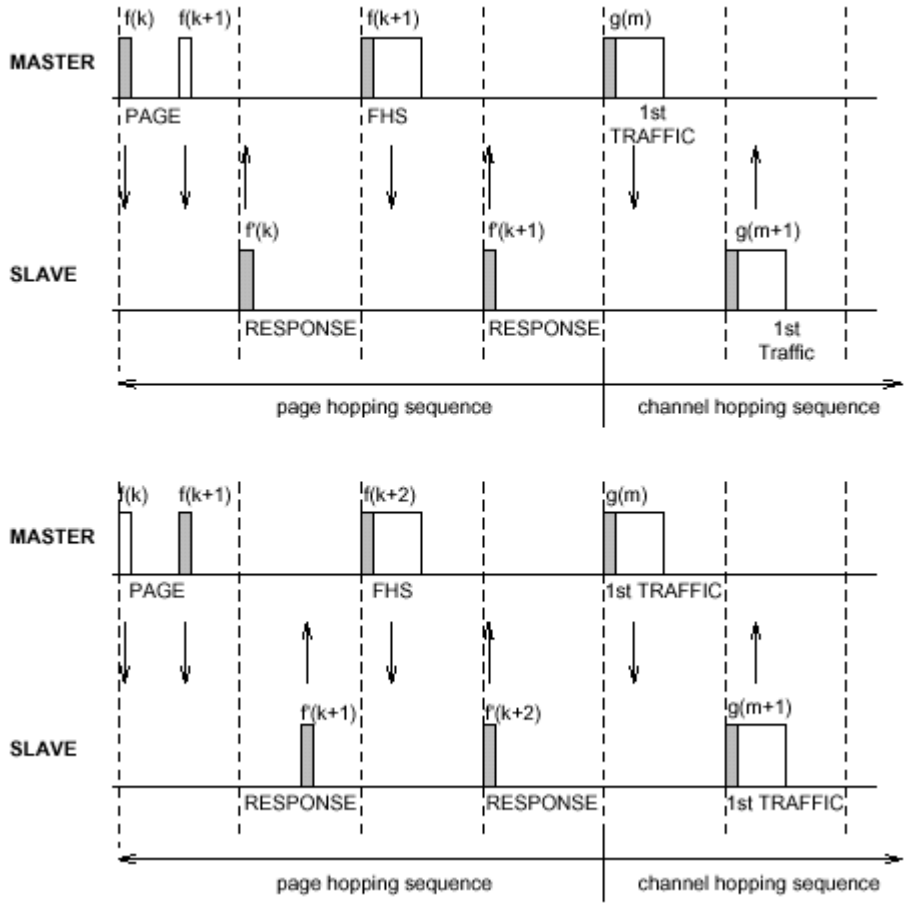
Periodicamente ogni unità si porta nel substato *page scan* e ascolta il canale per un tempo pari alla durata della scan window (tipicamente 18 slot ovvero 11.25 ms) per vedere se qualcuno sta trasmettendo il suo device access code. Durante la scan window l'unità si mette in ascolto su una frequenza selezionata dalla sequenza di page hopping (sequenza di 32 frequenze determinata dall'indirizzo BD_ADDR dell'unità) in base ai bit 16-12 del clock nativo dell'unità e quindi viene scelta una nuova frequenza ogni 1.28s . L'intervallo di scan, ovvero l'intervallo tra l'inizio di due scan window consecutive, può essere pari alla durata di una scan window (ovvero scan continuo) oppure 1.28s oppure 2.56s . Se l'unità si porta nel substato *page scan* dalla stato *connection* può accadere che lo scan sia interrotto dagli slot riservati a collegamenti SCO (che hanno priorità

maggiore) mentre collegamenti ACL possono essere posti in modalità hold o park. Un'unità che vuole stabilire una connessione (e diventare quindi master) si porta nel substato *page* in cui trasmette ripetutamente il device access code dell'unità che vuole contattare. Il master però non sa quando l'unità si mette in ascolto e su quale frequenza. Dalla conoscenza dell'indirizzo dell'unità il master determina la sequenza di page hopping dell'unità mentre si fa un'idea della fase effettuando una stima del clock nativo dell'unità utilizzando informazioni ottenute in un precedente incontro o con una procedura di inquiry. Siccome il pacchetto ID contenente il DAC dell'unità è lungo solo 68 bit il master può trasmetterlo su 2 diverse frequenze in uno stesso slot di trasmissione e nel seguente slot di ricezione può mettersi in ascolto sequenzialmente sulle 2 frequenze corrispondenti appartenenti alla sequenza di page-response hopping. Le 32 frequenze della sequenza di page hopping sono suddivise in due treni, A e B, di 16 frequenze ciascuno (A contiene le frequenze nell'intorno di quella stimata). Il master può trasmettere le frequenze di un treno in 16 slot ovvero 10 ms e ripete la trasmissione di questo treno un fissato numero di volte, a meno che non ci sia risposta dell'unità. Se non si è ottenuta risposta si prova a trasmettere ripetutamente il treno B, dopodichè di nuovo il treno A. La trasmissione alternata dei treni A e B è ripetuta finchè non si riceve una risposta o scade un timer. I pacchetti SCO hanno la precedenza rispetto ai pacchetti ID di paging ma non influenzano le frequenze negli slot non riservati :



Quando lo slave riconosce il proprio device access code passa nel substato *slave response* e invia il proprio DAC 625μs dopo l'inizio del pacchetto ID ricevuto utilizzando la frequenza corrispondente a quella alla quale ha ricevuto il messaggio di paging, come mostrato in figura. Dopo 312.5 μs attiva il ricevitore e si mette in ascolto. Quando il master riceve il DAC dello slave passa nel substato *master response* e invia, all'inizio del seguente slot master-to-slave, un pacchetto FHS che contiene tutte le informazioni per costruire il channel access code. Notiamo che il campo access code del pacchetto FHS contiene ancora il device access code dello slave e la frequenza utilizzata appartiene alla sequenza di page hopping. Quando lo slave riceve il pacchetto FHS risponde con il proprio DAC, calcola il channel access code e la sequenza di channel hopping, aggiunge un offset al proprio clock nativo per sincronizzarsi con quello del master e finalmente passa allo stato *connection*. Se il master riceve questa risposta passa anch'esso allo stato *connection* altrimenti

ritrasmette il pacchetto FHS. Nello stato *connection* si usa il channel access code e la sequenza di channel hopping.



a) lo slave risponde al primo messaggio di page

b) lo slave risponde al secondo messaggio di page

Il primo pacchetto trasmesso dal master nello stato *connection* è un pacchetto POLL per verificare che lo slave stia adottando la giusta frequenza di channel hopping; lo slave deve rispondere con un pacchetto qualsiasi. Per quanto riguarda la tempificazione, il master trasmette Mx1250µs dopo l'inizio della sua precedente trasmissione (M dipende dai pacchetti trasmessi e ricevuti) mentre lo slave trasmette Nx625µs dopo l'inizio della sua precedente ricezione (N dipende dai pacchetti ricevuti).

4.2.2 Procedura di inquiry

La procedura di paging presuppone che il master conosca l'indirizzo dello slave. La procedura di inquiry serve a scoprire quali unità Bluetooth sono presenti, il loro indirizzo e il loro clock. Il messaggio di inquiry non contiene alcuna informazione sul mittente ma solo il campo access code, di tipo GIAC o DIAC. Una unità che consente di essere scoperta passa periodicamente nel substato

inquiry scan, molto simile al substato *page scan*, in cui ascolta il canale per vedere se qualcuno sta trasmettendo un inquiry access code. Durante una scan window viene utilizzata una frequenza selezionata tra le 32 della sequenza di inquiry hopping, che è sempre determinata dal LAP del general inquiry access code (GIAC). La fase dipende dal clock nativo dell'unità che effettua lo scan e cambia ogni 1.28s. Se riceve un messaggio di inquiry l'unità può rispondere o meno; la risposta consiste in un pacchetto FHS, contenente il proprio device access code. Se più unità Bluetooth rispondono contemporaneamente si può verificare una collisione; anche se questo evento è abbastanza improbabile (nello stesso istante più unità devono essere in ascolto sulla stessa frequenza, la quale dipende dal clock nativo) è utilizzato il seguente protocollo. Quando un'unità riceve un messaggio di inquiry genera un numero casuale RAND compreso tra 0 e 1023 e ritorna nello stato *connection* o *standby* per un tempo pari a RAND time slot. Trascorso questo tempo torna nel substato *inquiry response* e al primo messaggio di inquiry ricevuto risponde con un pacchetto FHS e ritorna nello stato *inquiry scan* (quindi successivamente può rispondere di nuovo, a frequenze diverse). I pacchetti SCO hanno la precedenza anche sui messaggi di inquiry, quindi se un pacchetto di risposta si presenta in uno slot riservato a pacchetti SCO non viene spedito ma si aspetta il successivo messaggio di inquiry. Un'unità che vuole scoprire le altre unità che la circondano si porta nel substato *inquiry* e trasmette ripetutamente il messaggio di inquiry, utilizzando due treni (A e B) di 16 frequenze ciascuno, proprio come nel substato *page*. Tra una trasmissione e l'altra si mette in ascolto per ricevere eventuali pacchetti FHS, ai quali peraltro non risponde ma continua a trasmettere messaggi di inquiry e ad ascoltare eventuali risposte. Questa fase continua finchè l'unità non decide di aver acquisito sufficienti informazioni o scade un timer.

Un' unità Bluetooth che si trova nello stato *connection* può operare in 4 diverse modalità di funzionamento :

- Active mode : l'unità partecipa attivamente trasmettendo e ricevendo pacchetti; nel master-to-slave slot ascolta il pacchetto che viene trasmesso sul canale : se è ad essa indirizzato può rispondere nel seguente slave-to-master slot, altrimenti può dormire fino alla successiva trasmissione del master. E' richiesta una trasmissione periodica del master per mantenere gli slave sincronizzati al canale; siccome è necessario il channel access code un qualunque pacchetto è sufficiente
- Sniff mode : lo slave non è in ascolto in tutti gli slot master-to-slave ma soltanto in alcuni, regolarmente spaziati, in cui ascolta a chi è indirizzato il pacchetto; se è ad esso indirizzato continua ad ascoltare. Per attivare questa modalità il master utilizza un messaggio LMP

- Hold mode : lo slave non riceve pacchetti ACL (può ricevere pacchetti SCO) e può portarsi nei substati *page/inquiry scan, page, inquiry*; la durata di questa modalità è concordata tra il master e lo slave, che comunque mantiene il suo active member address (AM_ADDR)
- Park mode : quando uno slave non deve usare il canale della piconet ma vuole restare sincronizzato può passare a questa modalità, che è una modalità a basso consumo. Lo slave rilascia il suo AM_ADDR ma acquisisce due nuovi indirizzi di 8 bit : Parked Member Address (PM_ADDR, usato per procedure di unpark iniziate dal master) e Access Request Address (AR_ADDR, usato per procedure di unpark iniziate dallo slave). Quando ci sono unità in park mode il master stabilisce un canale di segnalazione (beacon channel) che consiste in un treno di beacon slot equidistanti trasmessi ad intervalli regolari e le cui funzioni sono : trasmissione di pacchetti master-to-slave alle unità in park mode da usare per la sincronizzazione, trasporto di messaggi per cambiare i parametri del canale di beacon, trasmissione di pacchetti generali di broadcast alle unità in park mode, consentire procedure di unparking. E' inoltre definita una finestra d'accesso in cui gli slave in park mode possono richiedere una procedura di unpark. Questa modalità consente di connettere più di 7 slave ad un singolo master e di funzionare in regime di basso consumo.

4.3 Scatternet

Diverse piconet possono ricoprire la stessa area. Siccome ciascuna piconet ha il proprio master le piconet hanno diverse sequenze di channel hopping e i pacchetti spediti sui loro canali hanno diversi channel access code. All'aumentare delle piconet ovviamente aumenta la probabilità di collisione e quindi la degradazione delle prestazioni. Un gruppo di piconet in cui esistono connessioni tra diverse piconet è detto scatternet. Un master o uno slave possono diventare uno slave in un'altra piconet se vengono contattati dal master di quella piconet; analogamente uno slave in una piconet può diventare master di un'altra piconet contattando altre unità. Ogni unità può partecipare a diverse piconet utilizzando la tecnica del time multiplexing, basta usare il channel access code e l'offset appropriati. Nel caso di collegamenti ACL un'unità può richiedere di passare in modalità hold o park in una piconet per unirsi ad un'altra piconet; le unità in sniff mode possono avere tempo sufficiente per visitare un'altra piconet tra due sniff slot. Se sono stati stabiliti collegamenti SCO si possono visitare altre piconet solo negli slot non riservati (in pratica ciò è possibile solo nel caso di un singolo collegamento SCO che utilizza pacchetti HV3).
E' previsto inoltre uno scambio di ruoli tra il master e uno slave, che comporta una ridefinizione della piconet in quanto i parametri di una piconet derivano dall'indirizzo e dal clock del master.

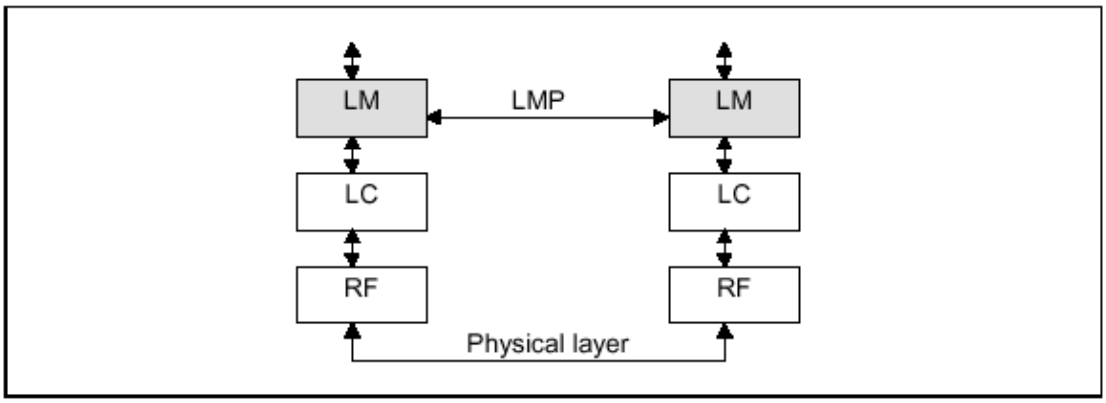
Uno dei principali obiettivi di Bluetooth è sicuramente quello di ridurre i consumi. Diversi accorgimenti sono stati pensati a tale scopo. Per esempio, se si devono inviare informazioni di controllo non si trasmette alcun payload; in ogni caso si trasmettono solo i bit utili. In ricezione se si trova un access code non valido o fallisce l'HEC o il pacchetto è indirizzato ad altri, l'unità torna a dormire. Inoltre ci sono modalità che riducono i consumi ovvero, in ordine decrescente di duty cycle, sniff mode, hold mode e park mode.

5. Audio

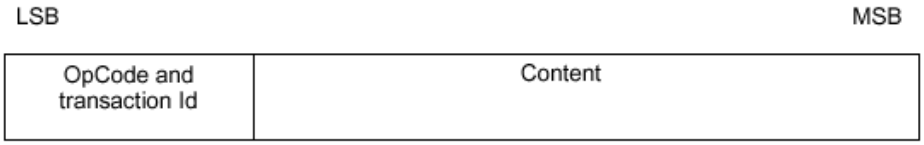
La funzione di questo strato è quella di codificare il segnale audio. Due tecniche possono essere adottate : log PCM e CVSD; entrambe forniscono un flusso di bit a 64 kbps.
La codifica log PCM (Pulse Code Modulation) consiste in una quantizzazione non uniforme a 8 bit che usa un compressore avente caratteristica logaritmica (A-law o μ-law) e nella codifica del valore quantizzato in impulsi 0-1. Nella codifica CVSD (Continuous Variable Slope Delta Modulation) il bit d'uscita indica se il valore predetto è maggiore o minore del valore della forma d'onda in ingresso, costituita da un segnale PCM con quantizzazione uniforme. Il passo è determinato dalla pendenza della forma d'onda.
Diciamo qualcosa sulla gestione di errori. Nei pacchetti DV e HV3 il payload non è protetto da codifica FEC quindi la qualità della voce dipende dallo schema di codifica adottato. La codifica CSVD si mostra più robusta, cioè è piuttosto insensibile a bit errati casuali. Nei pacchetti HV2 si usa una codifica FEC 2/3; errori non correggibili dovrebbero essere ignorati, nel senso che la stringa decodificata dovrebbe comunque essere passata al livello superiore. Nei pacchetti HV1 si usa una codifica FEC 1/3; si prende una decisione "a maggioranza" e quindi non si presentano errori non corretti.

6. Link Manager Protocol (LMP)

Il protocollo LMP permette la comunicazione tra i livelli Link Manager dei dispositivi.



I messaggi vengono spediti nel payload dei pacchetti ACL e si distinguono per il valore 11 assunto dal campo L_CH nell'header. Il formato della PDU è il seguente:

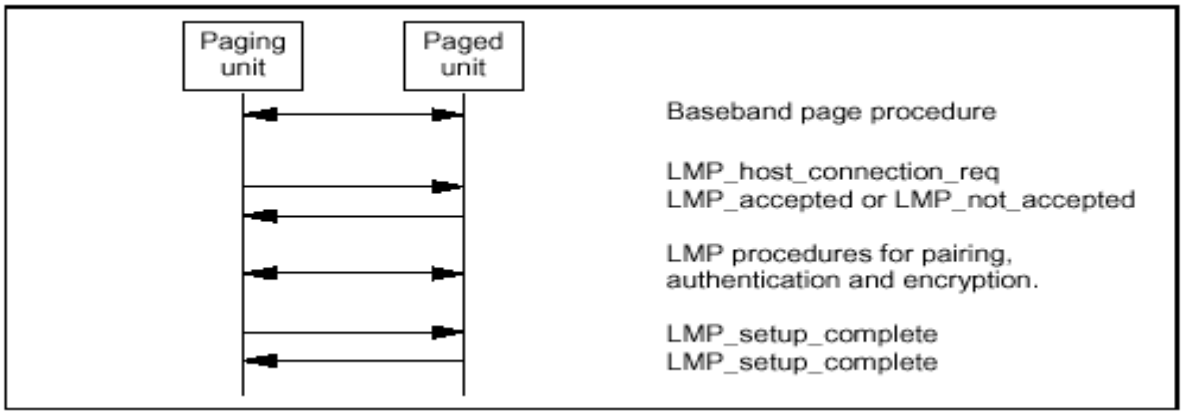


L'OpCode permette di distinguere i vari messaggi LM, il transaction Id è invece un bit che permette di stabilire se il colloquio è stato iniziato dal master o dallo slave. Infine nel campo Content si pongono gli eventuali parametri che caratterizzano il messaggio.
Il compito del livello Link Manager è di creare e gestire il collegamento tra le varie unità. Scendendo più in dettaglio possiamo dire che le funzionalità offerte sono:

- Gestione della piconet
  - Creazione (e chiusura) di collegamenti ACL o SCO
  - Cambio di ruolo (master-slave)
  - Gestione dei modi di funzionamento a bassa potenza (Hold, Sniff, Park)
- Configurazione del Link
  - Funzioni supportate, tipi di pacchetto validi
  - Qualità del servizio
  - Controllo della potenza
- Gestione della security
  - Autenticazione
  - Crittografia e gestione delle "chiavi"

6.1 Gestione della piconet

Come mostrato a livello BaseBand un collegamento viene avviato tramite operazioni di paging o di inquiry, dopo questa fase il link vero e proprio viene creato grazie ad una comunicazione a livello LM che possiamo tradurre in questo modo:



Si nota una fase di richiesta di connessione da parte dell'host che può essere accettata o meno, seguono poi le procedure di pairing, autenticazione e criptazione che affronteremo in dettaglio in seguito. A questo punto non sono necessarie altre operazioni e la fase di setup del link viene dichiarato completa.

In realtà il link creato in questo modo è di tipo ACL (Asynchronous Connection-Less), a questo punto uno o più link SCO (Synchronous Connection-Oriented) possono essere creati dal master o dallo slave grazie all'invio di un pacchetto che contenga i parametri del link (slot riservato e delay). In modo del tutto analogo le unità possono richiedere di cambiare i parametri del link o di chiuderlo.

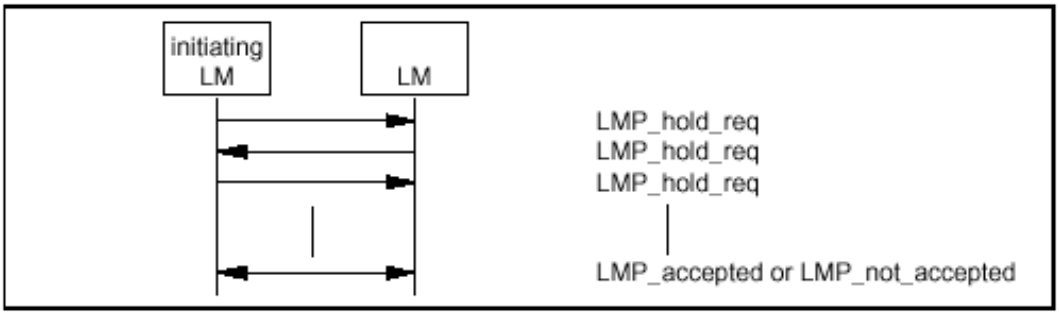
Invece la chiusura definitiva del collegamento (quindi del link ACL) può avvenire in un qualsiasi momento ad opera del master o dello slave con un pacchetto detto *LMP_detach.*

Consideriamo ora il cambio di ruolo, esistono alcune situazioni in cui può essere richiesto uno scambio di ruoli tra master e slave, questa procedura viene avviata grazie allo scambio di un opportuno pacchetto LMP. Ovviamente ci sarà un'unità che effettuerà la richiesta di questo scambio e l'altra unità interessata potrà accettare o meno. Se viene trovato l'accordo verrà avviata la vera e propria procedura di scambio.

Come anticipato a livello LM avviene anche la gestione dei modi di funzionamento a bassa potenza che ricordiamo essere le modalità di *Hold*, *Sniff* e *Park.*

Quando le unità sanno che non avranno dati da scambiarsi per un periodo relativamente lungo possono entrare in modalità Hold, durante questa fase il link ACL viene in pratica congelato e le

due device possono spegnere i loro transceiver. Sia il master che lo slave possono richiedere di entrare in hold mode tramite un pacchetto detto *LMP_hold_req*, nel quale viene indicato anche un hold-time indicativo, a questo punto può iniziare una vera e propria negoziazione tra le due unità al fine di determinare un hold time che vada bene ad entrambe.



Tale modalità può essere anche forzata da una delle due unità a patto che in precedenza fosse già stata accettata, l'hold-time indicato nel pacchetto (LMP_hold) non potrà comunque essere superiore a quello che era stato concordato in precedenza.

In modalità Sniff invece in effetti il link è attivo solo per una frazione di tempo, il master allora potrà trasmettere solo per questa frazione di tempo. La procedura si svolge in modo analogo a quanto visto sopra, quindi inizialmente ci sarà una negoziazione tramite i pacchetti *LMP_sniff_req*, che conterranno parametri come il periodo della fase di Sniff e la sua durata (in termini di slot). Sempre in analogia a quanto accade per la modalità Hold il master (ma non lo slave in questo caso) può forzare la commutazione.

Facciamo osservare che nel caso dell'Hold viene specificato un hold-time trascorso il quale tale modalità termina, nel caso dello Sniff invece un tale parametro non esiste quindi per terminare tale fase viene spedito il pacchetto *LMP_unsniff_req* (tale richiesta è poi soggetta all'approvazione o meno da parte dell'altra unità). Se la richiesta viene accolta il link viene riportato nella modalità active.

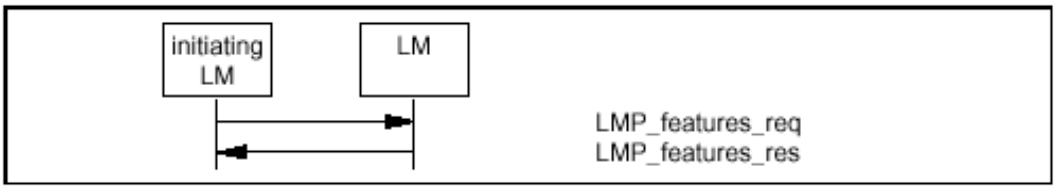
Ci resta infine da valutare la modalità Park; un'unità che si trovi in Park mode cede l'indirizzo che l'identifica nella piconet ma comunque continua a sincronizzarsi "risvegliandosi" ad intervalli prestabiliti. Questa modalità tipicamente viene attivata in due casi :

1) Il master deve garantire che il numero di slave all'interno della piconet sia pari al massimo a 7, quindi può forzare (*LMP_park*) o richiedere (*LMP_park_req*) ad uno slave di entrare in Park mode (in questo caso lo slave può anche rifiutare)
2) Lo slave vuole consumare meno energia per questo motivo richiede al master di porlo in modalità Park (il master può anche non accettare tale richiesta).

Il master può riattivare le unità poste in modalità park inviando un pacchetto in broadcast, in tale pacchetto esso indica gli indirizzi (BD_ADDR) delle unità che devono riattivarsi e gli assegna un nuovo indirizzo nella picocella (AM_ADDR).

6.2 Configurazione del link

Non è detto che una device Bluetooth supporti tutte le funzioni o tutti i tipi di pacchetti specificati nello standard, per questo motivo viene data la possibilità di chiedere all'interlocutore quali sono le funzioni che supporta. La procedura è molto semplice e si può riassumere con questa figura:



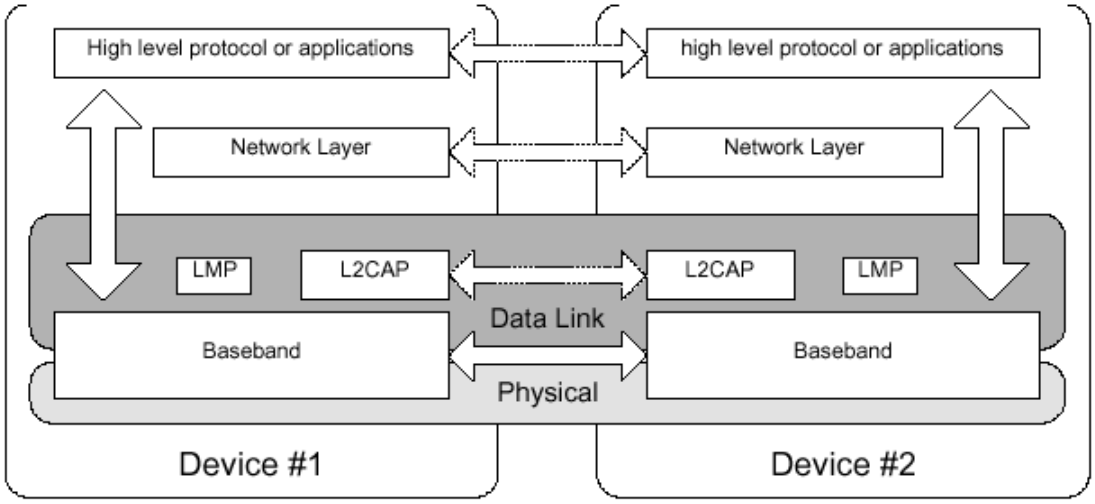
Un link è anche caratterizzato da una qualità del servizio misurata tramite il parametro detto poll interval, esso è il tempo massimo che intercorre tra due trasmissioni successive del master su questo link (chiaramente per il restante tempo il master colloquierà con gli altri slave). Il master può comunicare allo slave quale è il poll interval che gli garantisce, inoltre lo slave può richiedere una certa QoS ed il master deciderà se accettare o rifiutare tale richiesta, in questo modo si può creare una negoziazione dinamica della QoS del link.
La configurazione del link viene completata dalle funzioni di controllo della potenza trasmessa, in pratica se una device riscontra che la potenza che riceve dal suo interlocutore si discosta troppo dal valore ottimo può chiedergli di alzare o abbassare la potenza trasmessa. Le variazioni vengono eseguite per step fissati e quindi potrebbero essere necessari più messaggi prima di raggiungere il valore ottimale, inoltre la potenza potrebbe già essere al massimo (minimo), in tal caso questo viene comunicato e quindi potrà essere richiesta un aumento (diminuzione) di potenza solo in seguito ad una diminuzione (aumento) della stessa. Risulta evidente inoltre che ogni slave relativo ad uno stesso master in generale riceverà una potenza diversa, e quindi il master dovrà trasmettere con potenze diverse verso slave diversi.

6.3 Gestione della security

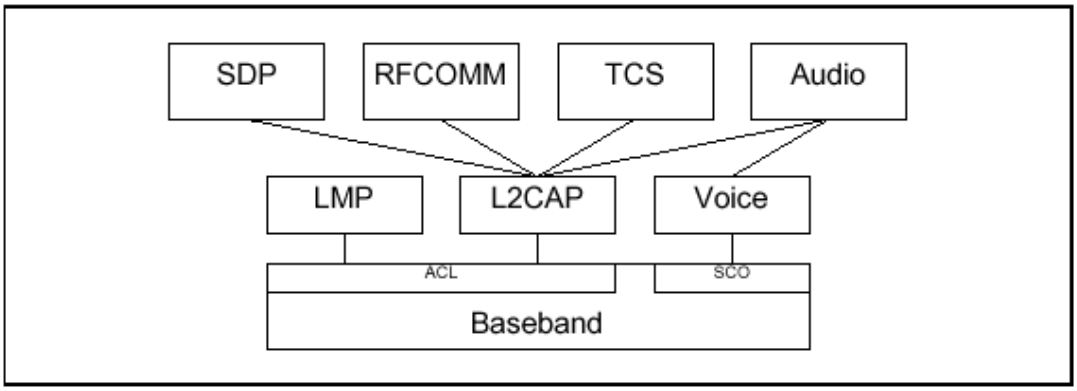
Al fine di garantire la sicurezza dei collegamenti sono state previste l'autenticazione (riconoscimento delle unità) e la crittografia dei dati. Al livello LM vengono scambiati i messaggi necessari per la creazione della link key che poi viene utilizzata nella fase di che consiste nell'invio del numero random e relativa risposta calcolata grazie alla conoscenza della link key.
Per quanto riguarda invece la crittografia a livello LM viene stabilita la modalità (crittografia attivata o meno, crittografia attivata solo sui pacchetti punto-punto o anche sui pacchetti broadcast), inoltre viene concordata la lunghezza della chiave.

7. L2CAP

Lo strato L2CAP è posizionato all'altezza del livello DLC dello stack Bluetooth parallelamente allo strato di signalling e controllo LMP.



Esso giace al di sopra del livello Baseband, del quale sfrutta i servizi, ed è stato implementato al fine di garantire il *trasferimento dei dati*, in modalità sia Best Effort che Connection oriented, dei livelli superiori inferfaccianti con lo stesso (SDP, RFCOMM, TCS, PACKETIZED AUDIO).



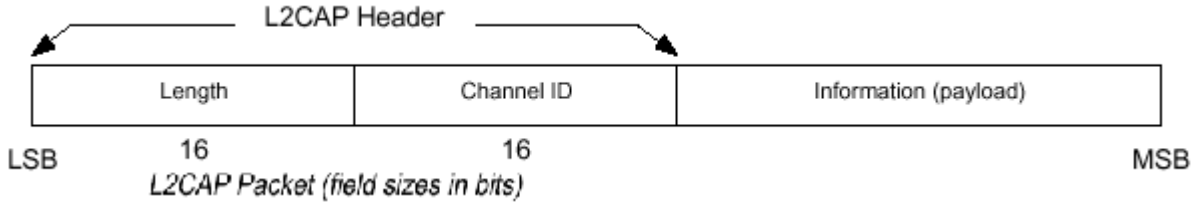
L2CAP, sostanzialmente, permette di trasmettere e ricevere pacchetti di dimensioni fino a 64 KB ed utilizza il concetto di *canale*, nello stabilire quelli che sono i collegamenti tra più applicazioni operanti su device Bluetooth.

7.1 Funzionalità

Allo strato sono specificatamente demandate le funzionalità:

- di *multiplexing* dei dati appartenenti a differenti applicazioni
- la gestione dell'astrazione del *multicast*
- l'operazione di *Frammentazione–Riassemblaggio* dei PDU provenienti dal livello rete.

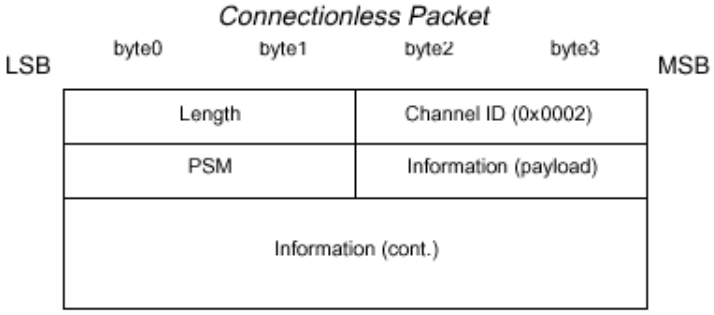
La comprensione di come tali operazioni siano realizzate è semplificata dall'analisi della struttura del generico pacchetto L2CAP che, come mostrato, si compone di tre campi



detti rispettivamente Lenght (16bit), CID (Channel Identifier) e Payload (dim max 65535 byte).

A riguardo è fondamentale osservare che la diversa specializzazione del campo CID consenta:

1) di individuare dinamicamente gli end point di un canale connection oriented (utilizzando valori per il campo compresi nell'intervallo (0x0040,0xFFFF))
2) di realizzare l'astrazione di un multicast best effort (CID fisso e pari 0x0002 sia in TX che RX) per ottenere la multiplazione del traffico relativo alle differenti applicazioni, che sono poi individuate in base al valore assunto dal campo PSM (nel caso parte iniziale del "payload")

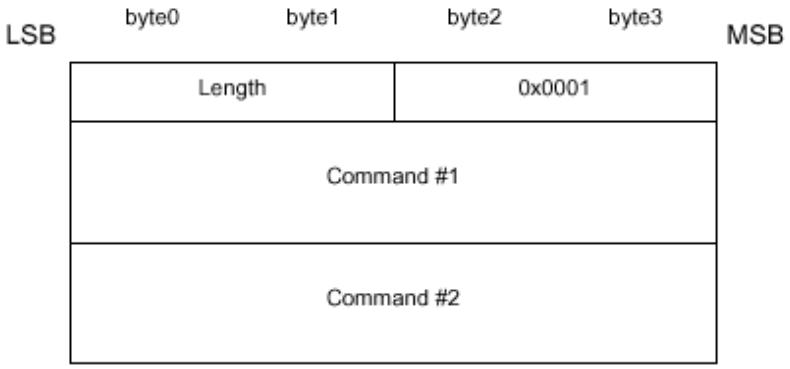


PSM value	Description
0x0001	Service Discovery Protocol
0x0003	RFCOMM
0x0005	Telephony Control Protocol
<0x1000	RESERVED
[0x1001-0xFFFF]	DYNAMICALLY ASSIGNED

3) di realizzare un canale di segnalazione tra entità pari, separato dal flusso dei dati (CID 0x0001).

7.2 Signalling e QoS

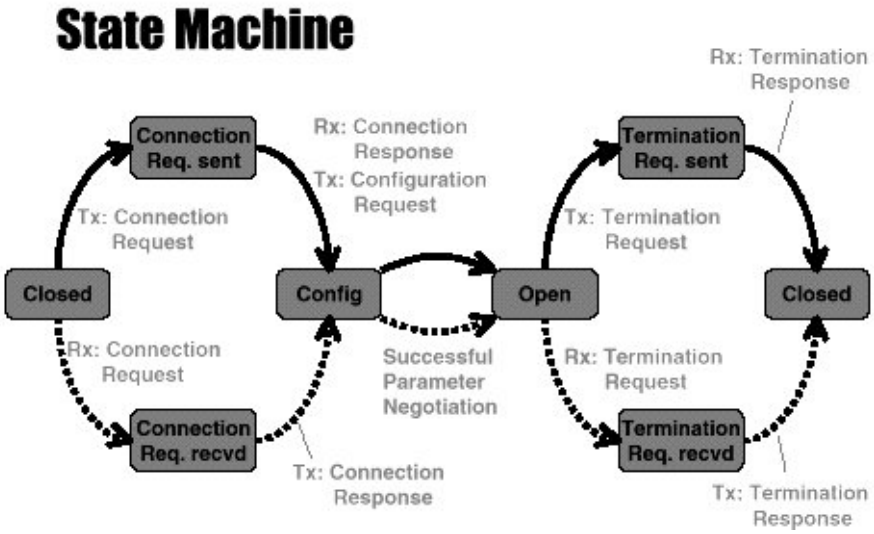
La struttura del pacchetto di signalling è la seguente:



Il signalling comprende quindi le operazioni di

- Richiesta di connessione
- Configurazione del canale
- Disconnessione

Facciamo notare come l'instaurazione di un canale, tra due entità pari corrispondenti a diverse device, si può inquadrare come passaggio sequenziale attraverso gli stati **Close-Config-Open-Close**

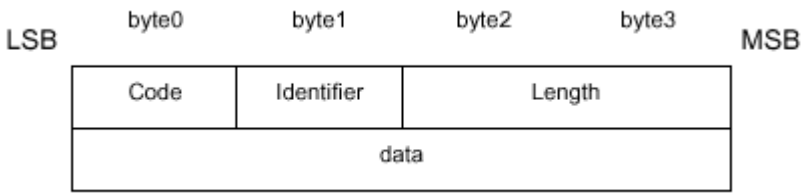


dell'automa di seguito presentato:

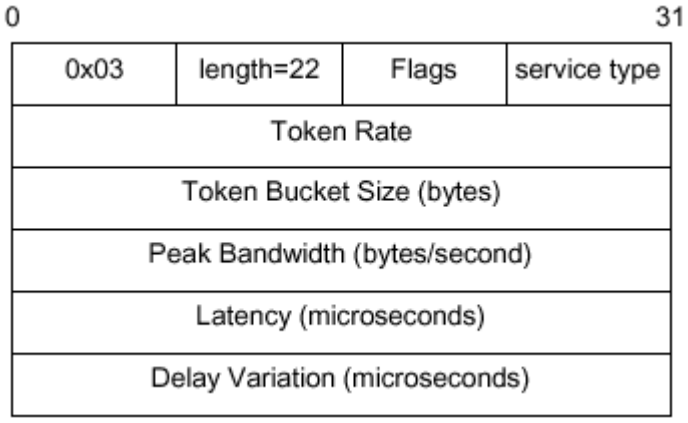
Il modello di connessione prevede la trasmissione (ricezione) di un pacchetto di Connection request (response), una eventuale fase di negoziazione (Configuration request/response packet) dei parametri della connessione, il colloquio, la chiusura del canale.
La fase di Configurazione è quella nell'ambito della quale si viene a definire la QoS della connessione e si esplicita nell'indicazione :

- dell' MTU (*Maximum transmission unit*) che si intende supportare (essendo il min fissato dalla struttura del generico pacchetto di signalling che è di 48 byte)
- nella specificazione del Flush Timeout Option che è parametro utilizzato per informare il ricevente del tempo limite che il trasmittente è intenzionato a spendere per trasmettere un pacchetto (porre il campo corrispondente a 111....111 implica il voler realizzare un canale affidabile, mentre porlo a 000...001 significa che si vuole adottare una soluzione best effort, che tra l'altro è l'unica specificabile nel caso di canali *point to multipoint* quindi multicast secondo quelle che sono le specificazioni dello strato L2CAP)

Le informazioni ora riportate sono contenute nel generico *campo comandi* dello specifico pacchetto di Signalling la cui struttura è di seguito rappresentata



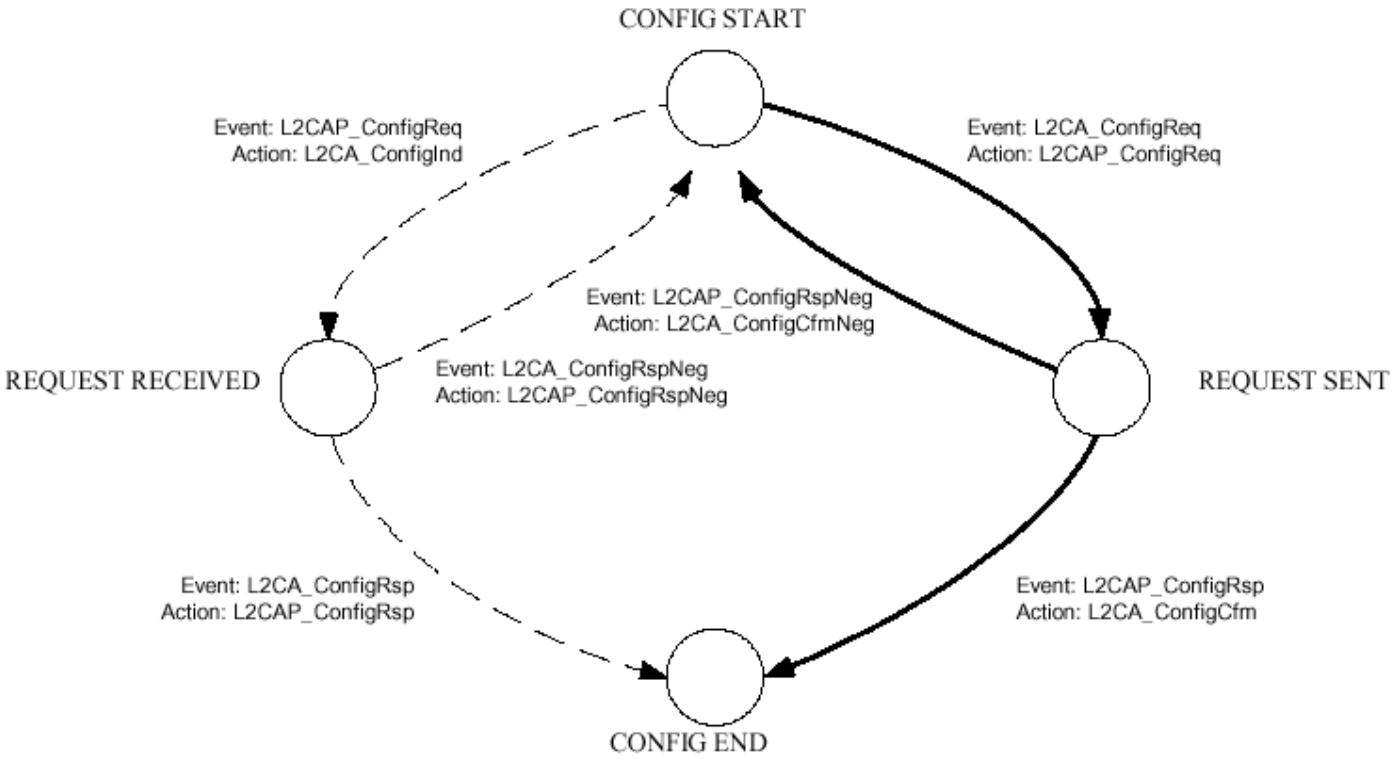
e nel quale sono specificabili differenti parametri



che corrispondono alla specializzazione del

- Livello di Servizio (Best effort / Guaranteed)
- Token rate (tasso garantito in bytes per secondo, da riguardarsi come *averange peak* consentito al trasmettitore): in particolare laddove il valore 0X000...000 implica assenza di specificazione (caso Best Effort) il valore 0XFFF...FFF è wild card al massimo disponibile.
- Token bucket: dimensionamento del buffer
- Latency: massimo tempo intercorribile tra la trasmissione di un bit dall'applicazione e il suo instradamento fisico nella piconet
- Delay Variation: Jitter in microsecondi (0XFFF...FFF significa *don't care*)

Non è superfluo ancora in questa fase sottolineare come il processo di configurazione del canale si realizzi attraverso le fasi di richiesta di specificazione del servizio e accettazione che sono poi da ripetersi due volte (perchè il canale offerto a livello Baseband è comunque full duplex, anche se il servizio può risultare logicamente simplex) e può essere astratto facendo riferimento alla figura



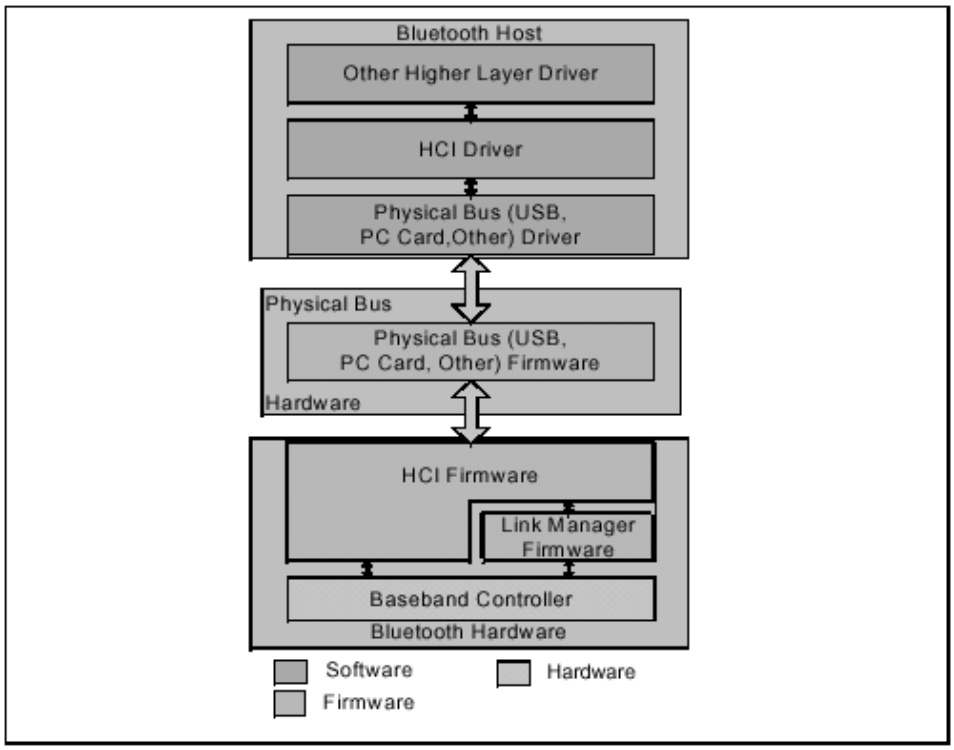
Come specifica dello strato L2CAP si è infine, ma non meno importante, evidenziata la capacità di Frammentazione e Riassemblamento delle unità dati provenienti dal livello superiore. In particolare è da sottolineate come il protocollo L2CAP non faccia altro che realizzare la frammentazione del flusso entrante dai livelli corrispondenti alle applicazioni (trattato 1:1) decomponendo i pacchetti ricevuti in spezzoni Baseband compatibili. L'esigenza di compiere una operazione di questo tipo corrisponde a quella di efficientizzare l'impiego della banda da parte delle applicazioni, i payload delle quali sono usualmente dimensionati in modo del tutto indipentente dal flusso effettivamente trasportato sul canale fisico che nel caso prevede pacchetti non ulteriori a 341 byte che limiterebbero drasticamente l'efficienza.

L2CAP realizza quindi, per specifica, su richiesta un canale affidabile per pacchetti utilizzando i meccanismi presenti a livello Baseband per il controllo della correttezza semantica del flusso (1 bit sequence number e 1 bit acknowledgement number) verificando semplicemente, a mezzo di un check di congruenza dimensionale, rispetto al campo Lenght, la correttezza del meccanismo di riassemblamento.

In pratica in fase di trasmissione gli spezzoni sono passati all' LMP a mezzo dell'Host Controller Interface e sono le informazioni introdotte dallo stesso a garantire il riaccorpamento dell'unità dati nell'ambito dell'usuale meccanismo di imbustamento multiplo del payload.

8. Host Controller Interface (HCI)

L' HCI è un'interfaccia verso il livello Baseband ed il Link Manager, questa interfaccia fornisce un metodo per accedere alle capacità del Baseband.



La figura mostra gli strati più bassi che caratterizzano una device Bluetooth, come si vede i livelli superiori comunicano con il BaseBand utilizzando i comandi messi a disposizione dall'HCI tramite l'HCI Driver. In effetti sarà poi l'HCI Firmware ad implementare tali comandi utilizzando le funzionalità messe a disposizione dal Baseband e dal Link Manager, nonché utilizzando i vari registri che sono presenti nella struttura hardware.

Sarà inoltre possibile utilizzare diversi tipi di bus fisico (detto anche HCI Transport Layer) , per il momento sono supportati i bus di tipo USB e PC Card.

I comandi messi a disposizione possono essere suddivisi in questo modo:

1) Comandi di controllo del link : Danno la possibilità all'host di controllare la connessione producendo messaggi LMP (ad esempio ci sono comandi per la creazione di link SCO, oppure per lo scambio delle chiavi etc.)
2) Comandi di policy del link : Sono usati per influenzare il modo in cui il LM gestisce la piconet (permettono ad esempio di gestire i modi di funzionamento a basso consumo)
3) Comandi informativi : Permettono di accedere a vari registri hardware al fine di trarne informazioni.

9. RFCOMM

Il protocollo RFCOMM fornisce per specifica l'emulazione di una seriale ed è posto al di sopra del protocollo L2CAP dello stack di Bluetooth.
Specificatamente si vanno ad emulare i nove circuiti propri dell'interfaccia RS-232 (EIATIA-232-E), dei quali nel seguito riportiamo i corrispondenti segnali di controllo dalle ben note funzionalità:

Pin	Circuit Name
102	Signal Common
103	Transmit Data (TD)
104	Received Data (RD)
105	Request to Send (RTS)
106	Clear to Send (CTS)
107	Data Set Ready (DSR)
108	Data Terminal Ready (DTR)
109	Data Carrier Detect (CD)
125	Ring Indicator (RI)

Table 2.1: Emulated RS-232 circuits in RFCOMM

Il protocollo prevede che una device Bluetooth possa supportare in una sessione (e in dipendenza dell'eventuale capacità) di lavoro più emulazioni di seriali (fino a 60) ciascuna individuata a mezzo del proprio DLC identifier (DLCI)

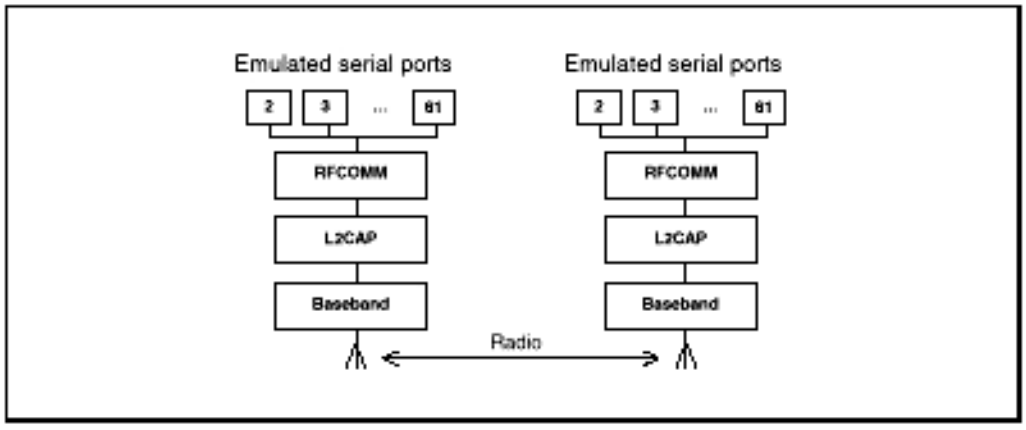


Figure 2.2: Multiple Emulated Serial Ports.

e nel caso gli end-point di alcune di queste seriali risultassero differenti (cioè la comunicazione coinvolga più di due device) non è superfluo sottolineare come distinti risulteranno i valori dei CID impiegati a livello L2CAP per il trasporto dei corrispondenti payload perché diversi i canali logici corrispondenti.

Al fine di capire la funzionalità dello strato software proprietario è utile inquadrare quella che è la funzione di RFCOMM nell'ambito di un tipico sistema che include l'impiego di una seriale:

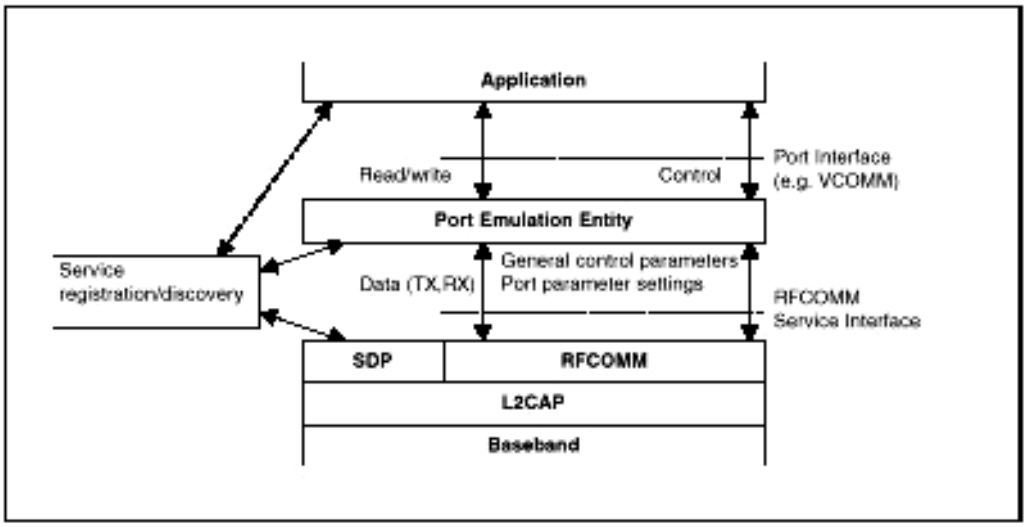
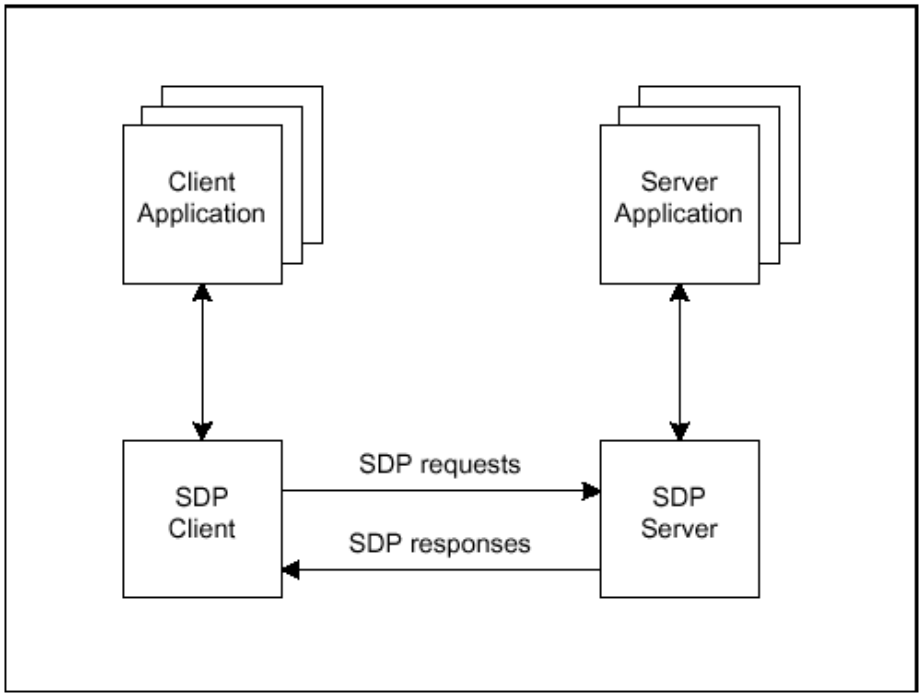


Figure 3.1: RFCOMM reference model

Il modello di riferimento presentato nelle specificazioni Bluetooth per RFCOMM suppone di aver a disposizione un' applicazione che utilizza un'interfaccia seriale proprietaria e prevede una "Port Emulation Entity" che mappa l'API corrispondente nei servizi RFCOMM di modo che l'applicazione con uscita seriale di riferimento possa utilizzare L2CAP per il trasporto punto-punto corrispondente.

10. Service Discovery Protocol (SDP)

Questo protocollo fornisce alle applicazioni un metodo per scoprire quali servizi sono disponibili e quali sono le loro caratteristiche, l'interazione è molto semplice ed avviene in questo modo:



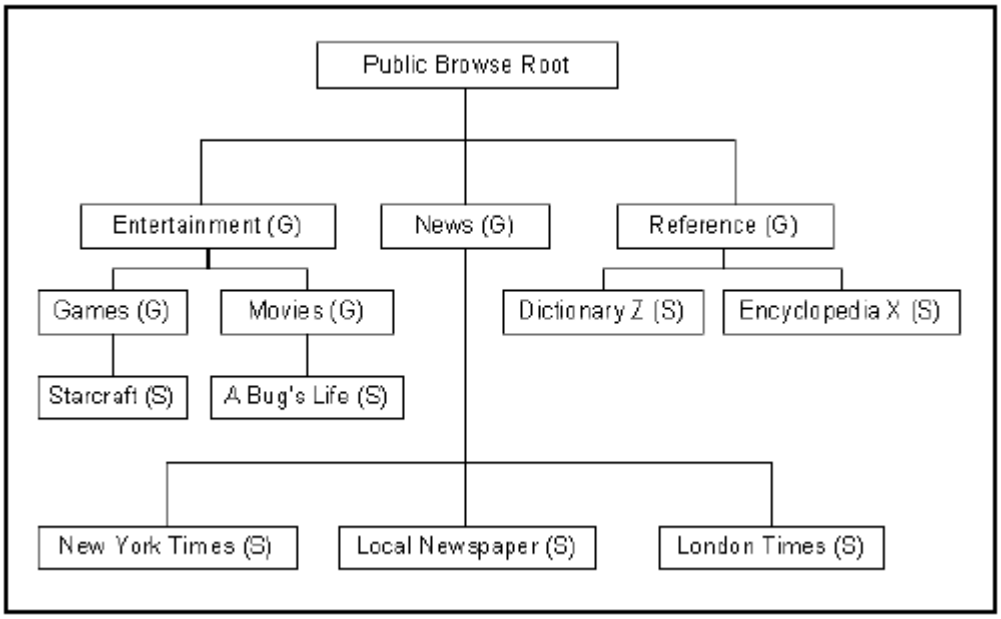
Quindi il protocollo entra in gioco nella comunicazione tra l'SDP client e l'SDP server.
E' bene precisare subito che SDP fornisce solo un modo per capire quali servizi sono disponibili in una determinata area, ma non permette di accedere agli stessi, se si vuole usufruire dei servizi offerti da un determinato server sarà necessario aprire una connessione verso quest'ultimo.
Va ricordato poi che una rete di dispositivi Bluetooth è estremamente dinamica (alcuni server possono uscire dal nostro range ed altri invece possono entrarvi in qualsiasi momento), quindi è stato pensato ad un meccanismo di notifica che interviene quando un nuovo server diviene disponibile, in questo modo il client può richiedere la lista dei servizi che esso offre. Quando invece un server lascia il range di funzionamento del client chiaramente non può esserci alcuna notifica, tuttavia il client tramite SDP può procedere ad un poll sui server e quindi risalire a quelli che eventualmente non sono più disponibili.
Come dicevamo prima, il server fornisce informazioni sui servizi che mette a disposizione, in effetti per ciascun servizio esso mantiene un Service Record nel quale sono memorizzati gli attributi del servizio. Ciascun attributo a sua volta è fatto da due campi che sono l'ID (identificatore dell'attributo) e il valore dell'attributo, esempi di attributi possono essere il nome del provider

oppure la classe alla quale il servizio appartiene (ad esempio stampa, possibili sottoclassi invece potrebbero essere la stampa a colori o in bianco e nero).

In effetti la ricerca dei servizi può avvenire in due modi diversi:

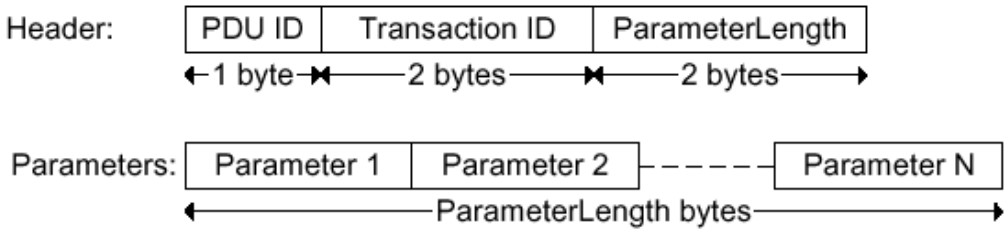
1) Il client sta cercando un servizio caratterizzato da determinati attributi e vuole sapere se il server è in grado di fornirgli un servizio che risponda a tali requisiti, in questo caso la ricerca avviene fornendo gli UUID (Universal Unique Identifier) di tali attributi.Gli UUID sono degli identificatori universalmente riconosciuti , e quindi non sono caratteristici del determinato server).
2) Il client non sta cercando un determinato servizio, ma vuole solo sapere quali sono i servizi che il server mette a disposizione, tale modalità è indicata con il nome di service browsing. La cosa ineteressante è che il server può organizzare i servizi che offre in gruppi caratterizzati da attributi simili, questo chiaramente rende molto più semplice il browsing.

In figura è mostrata una possibile organizzazione gerarchica dei servizi (con la lettera G sono indicati i gruppi, mentre con la lettera S i servizi veri e propri).



Ora scendiamo un po’ più nel dettaglio del protocollo.

I messaggi vengono trasportati dal livello L2CAP, il formato della PDU è il seguente:



Come vediamo nell'header c'è un campo PDU ID che identifica il tipo di messaggio, poi c'è un campo Transaction ID che identifica in modo univoco le richieste emesse dal client e permette allo stesso tempo quindi di stabilire a quali richieste si riferiscono le varie riposte dei server. C'è poi il campo ParameterLength che specifica la lunghezza in byte dei parametri, seguono poi i parametri appunto, essi contengono le eventuali informazioni specifiche del messaggio (ad esempio parametri per la ricerca, oppure in una risposta la lista dei servizi e i loro attributi.

11. TCS (Telephony Control protocol Specification Binary)

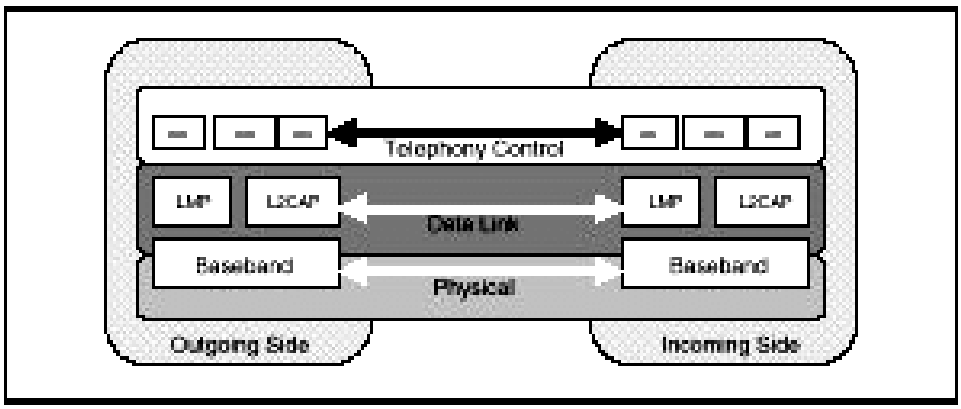


Figure 1.1: TCS within the Bluetooth stack

Lo strato TCS è basato sulla raccomandazione ITU-T Q.931 e contiene le seguenti funzionalità:

- Call Control (CC): corrispondente al signalling per la connessione e il rilascio di una chiamata vocale e/o dati tra due device Bluetooth
- Group Management: corrispondente al signalling per la gestione del multicast
- Connectionless TCS (CL): corrispondente al signalling per informazioni ulteriori tra device coinvolte nella chiamata

TCS utilizza una soluzione di signalling di tipo punto-punto quando è nota la device verso la quale la chiamata dovrà essere instradata (nel qual caso il protocollo di signalling è mappato in un canale connection oriented di L2CAP) ovvero di tipo punto-multipunto se esistono più device nella piconet verso le quali può essere instradata la chiamata (il signalling è allora mappato in un canale connectionless di L2CAP).

Nel primo caso al ricevente è notificata la richiesta di connessione usando un canale di signalling punto-punto (A), che è di seguito utilizzato ancora per definire il canale dati.

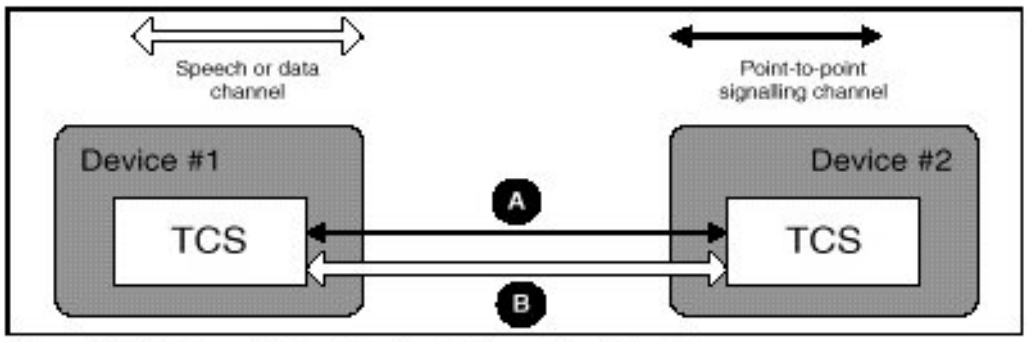


Figure 1.2: Point-to-point signalling in a single-point configuration

Nel secondo caso a tutte le device della piconet, attraverso un canale punto-multipunto (A) è comunicata la presenza di una chiamata in ingresso, e una delle device risponderà alla notifica attraverso un canale punto-punto che sarà utilizzato anche per la comunicazione vera e propria (B).

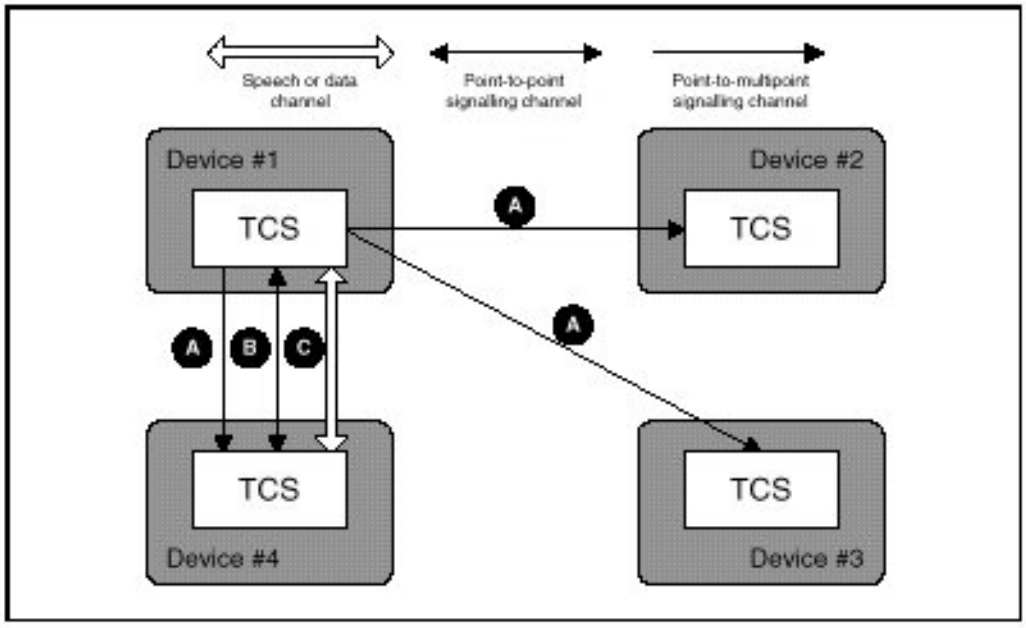


Figure 1.3: Signalling in a multi-point configuration

Le implementazioni dello strato TCS dovrebbero seguire la seguente architettura di riferimento per quel che riguarda le operazioni tra strati:

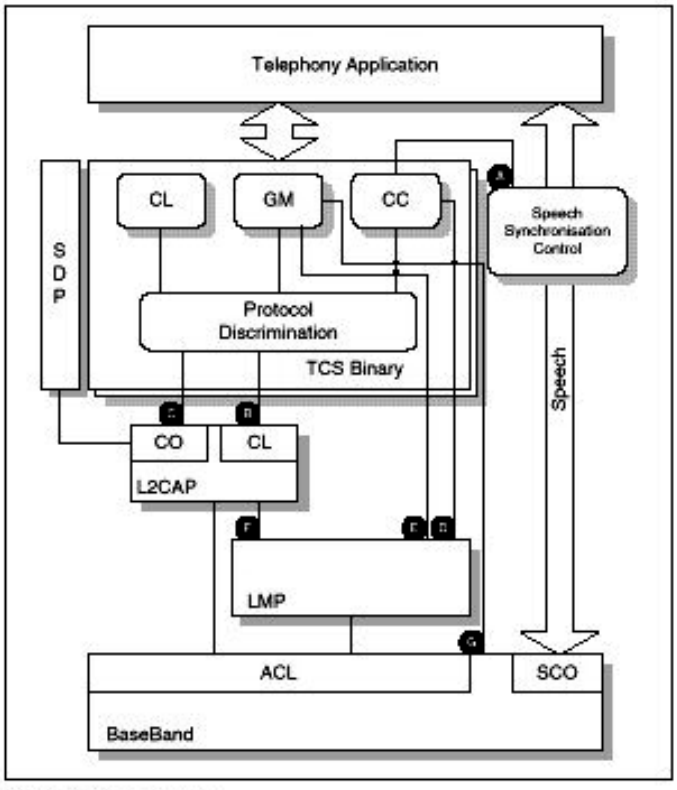


Figure 1.4: TCS Architecture

Nello spiegare la stessa evidenziamo come:

A) L'entità Call Control fornisce le informazioni relative alla sincronizzazione della chiamata nelle fasi di connessione e rilascio

B) Il setup, attraverso invocazione di tipo punto-multipunto, è instradato attraverso una soluzione L2CAP connectionless

C) Il setup, attraverso invocazione di tipo punto-punto è instradato attraverso una soluzione L2CAP connection oriented

D) L'entità Call Control controlla LMP in modo diretto per la connessione e il rilascio di canali si tipo sincrono

E & G) L'entità Group Management controlla LMP e Lc/Baseband in modo diretto durante le procedure di inizializzazione per il controllo delle fasi di inquiry, paging e pairing.

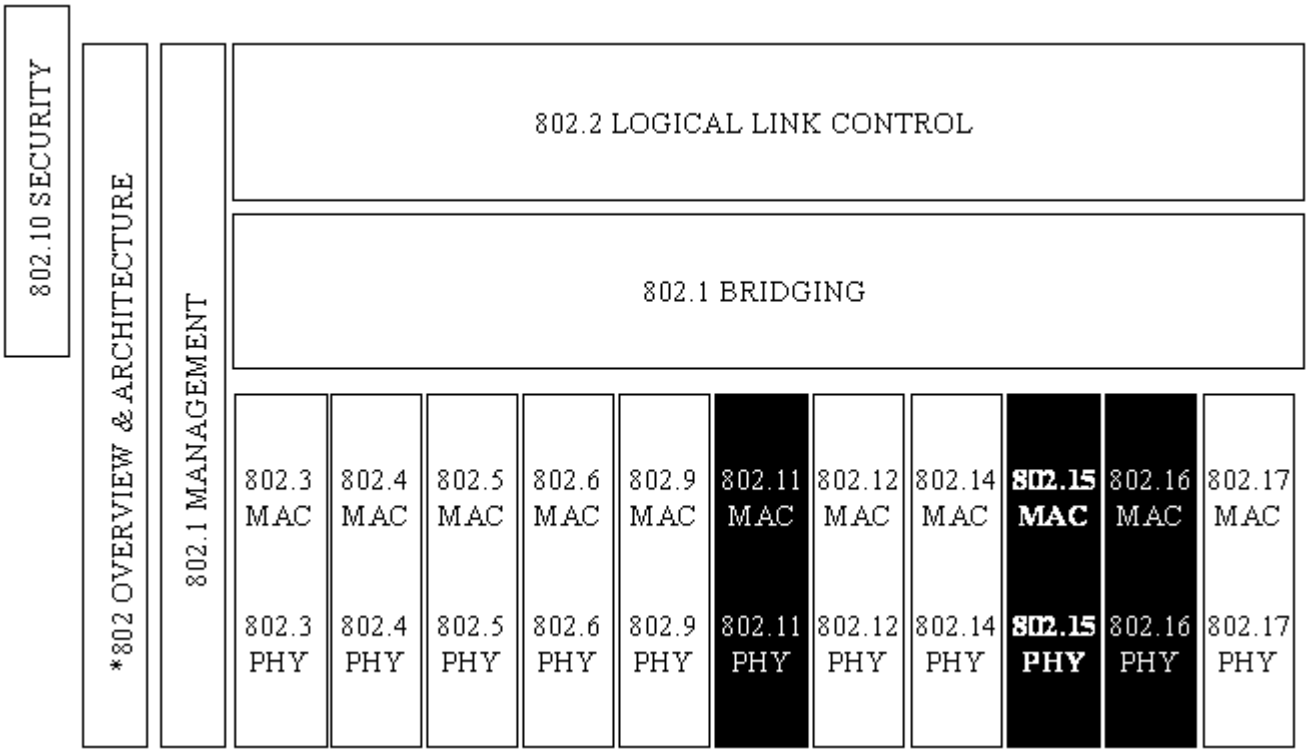
Si noti infine, come rappresentato, che le funzionalità dello strato TCS siano complementate a mezzo di un protocollo di discriminazione in base al quale è possibile instradare opportunamente il traffico verso le rispettive entità funzionali.

12. Lo standard 802.15

Il comitato IEEE 802 si occupa di definire e incoraggiare standard per i due livelli più bassi della pila OSI. Vari standard sono stati prodotti fino ad oggi, ricordiamo ad esempio 802.3-4-5 oppure il più recente 802.11 che standardizza le Wireless LAN.

Proprio dal comitato di 802.11 nasce il gruppo di studio sulle cosiddette WPAN (Wireless Personal Area Network), il cui scopo è di studiare la possibilità di realizzare delle reti wireless che colleghino tra di loro device anche molto diverse (come telefoni cellulari, stampanti, computer, per arrivare persino ad elettrodomestici). Una WPAN quindi è tipicamente una rete di piccole dimensioni che permette di eliminare tutti i collegamenti con cavi che affollano le case, l'aggettivo personali sta ad indicare che molto spesso le device in gioco sono di tipo personale e facili da trasportare "nel taschino". Le condizioni per la realizzabilità anche con le tecnologie attuali c'erano tutte e per questo motivo è stato avviato il processo di standardizzazione sotto il nome di 802.15.

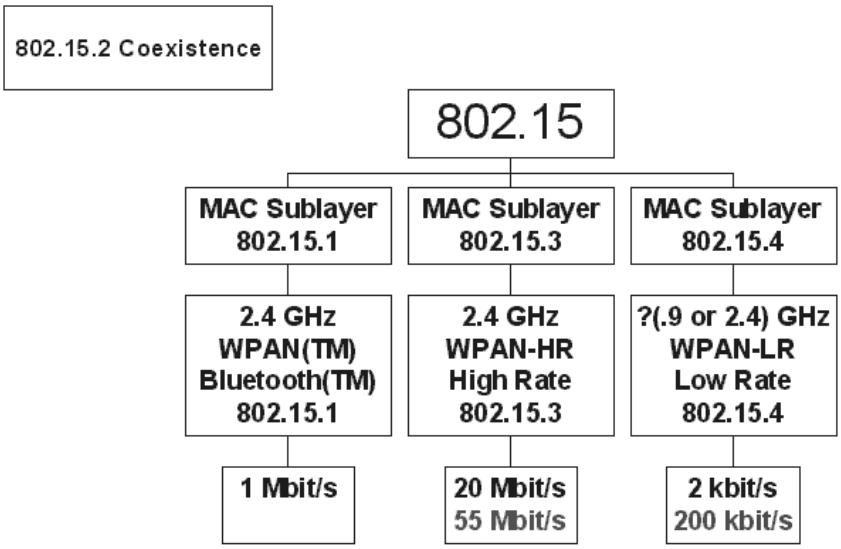
Il gruppo si è posto come obiettivo quello di produrre uno standard per la connettività senza filo tra device, mobili o fisse che siano, che utilizzi una tecnologia poco ingombrante economica e con bassi consumi. Un altro obiettivo ritenuto importante è di garantire l'interoperabilità con le reti 802.11. La seguente figura mostra come 802.15 si inserisce nel contesto degli altri standard 802.X



In realtà al momento della formazione di questo gruppo già alcune soluzioni si stavano presentando al pubblico, tra queste vanno ricordate Bluetooth e HomeRF (quest'ultima più orientata verso la creazione di una rete casalinga). Visto che entrambe le soluzioni sono sostenute da gruppi di aziende molto potenti (addirittura Bluetooth è appoggiata da un gruppo formato da circa 1900 aziende) e che sono molto promettenti da un punto di vista di diffusione sul mercato, il gruppo 802.15 sta cercando di produrre uno standard che sia compatibile con esse. In effetti è proprio Bluetooth quella che più si avvicina a soddisfare i requisiti richiesti per 802.15, questo ha comportato allora che lo standard 802.15 al momento attuale è praticamente identico a Bluetooth.
Per la precisione il gruppo di lavoro 802.15 è diviso in 5 sottogruppi che sono :

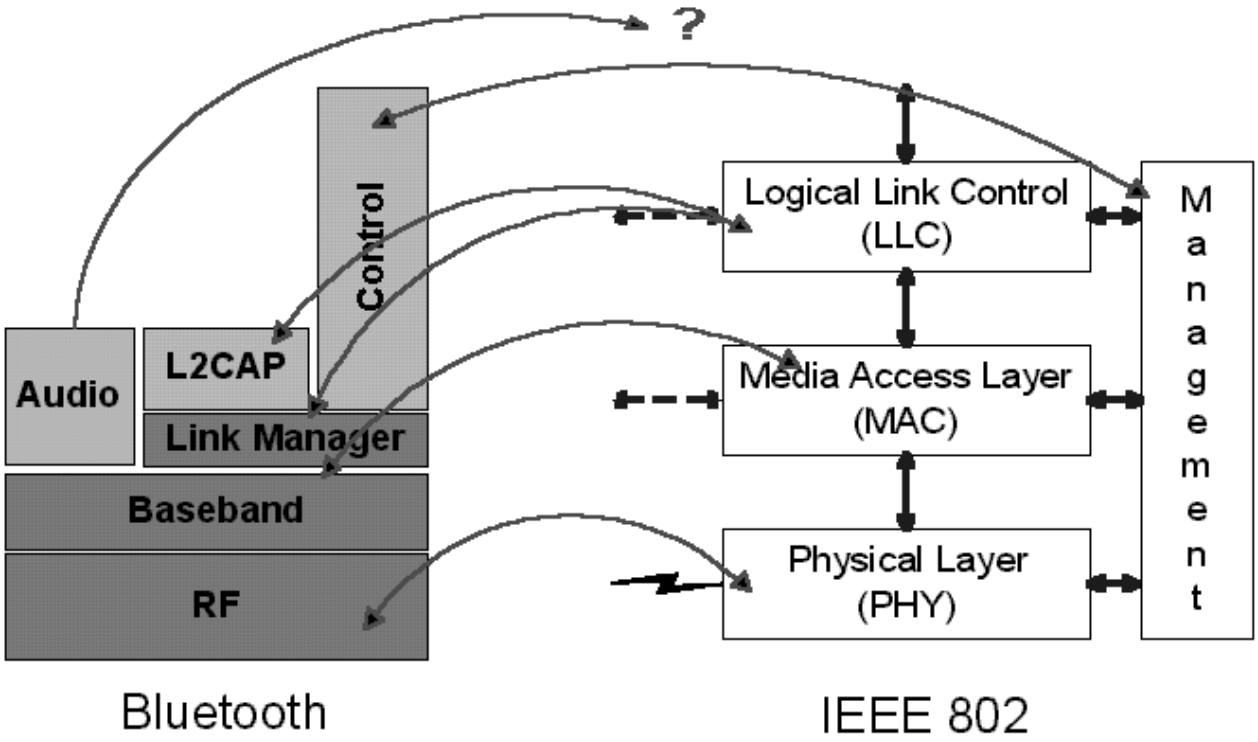
- TG1 (Task Group 1) , questo gruppo si sta occupando della derivazione di uno standard dalle specifiche di Bluetooth.
- TG2, che sta studiando la possibilità di far interoperare i dispositivi 802.15 con quelli 802.11
- TG3 che sta studiando la realizzabilità di uno standard ad alto tasso trasmissivo (20 Mbps o superiore) che mantenga però la compatibilità con lo standard prodotto dal TG1
- LRSG (Low Rate Study Group), questo gruppo sta studiando uno standard a basso tasso trasmissivo al fine di mantener ancora più contenuti il consumo, le dimensioni e la complessità.
- PC (Publicity Committee, il compito di questo gruppo è di assicurarsi che le proposte fatte dai vari gruppi di lavoro non si accavallino in modo da poter avere delle offerte sul mercato ben definite.

La gerarchia degli standard può essere quindi così riassunta:



Noi in particolare ci occuperemo ora brevemente del lavoro svolto dal TG1 (802.15.1) che è quello che ci interessa più da vicino. C'è da dire che il processo di standardizzazione è tuttora ancora in corso e dovrebbe concludersi il settembre prossimo, quindi tutto quello che verrà mostrato in seguito non necessariamente coinciderà con quello che poi sarà lo standard definitivo. Inoltre ci sono ancora dei dubbi su come risolvere alcuni problemi (come ad esempio la collocazione dello strato Audio)
Ovviamente lo sforzo principale è stato quello di cercare di mappare lo stack di protocolli di Bluetooth (o perlomeno quelli più bassi) sui primi due livelli di una pila OSI.

Nella figura che segue viene appunto mostrato in che modo si faranno corrispondere i livelli della pila Bluetooth con quelli della pila OSI.



Come era facile ipotizzare si è pensato ad una corrispondenza uno ad uno tra RF-PHY e BaseBand-MAC, inoltre è anche abbastanza intuitivo riconoscere che le funzionalità offerte dai due livelli L2CAP e LMP corrispondano con quelle del LLC, ancora non è chiaro invece in che modo si possa introdurre il livello Audio nella architettura 802.

Indice