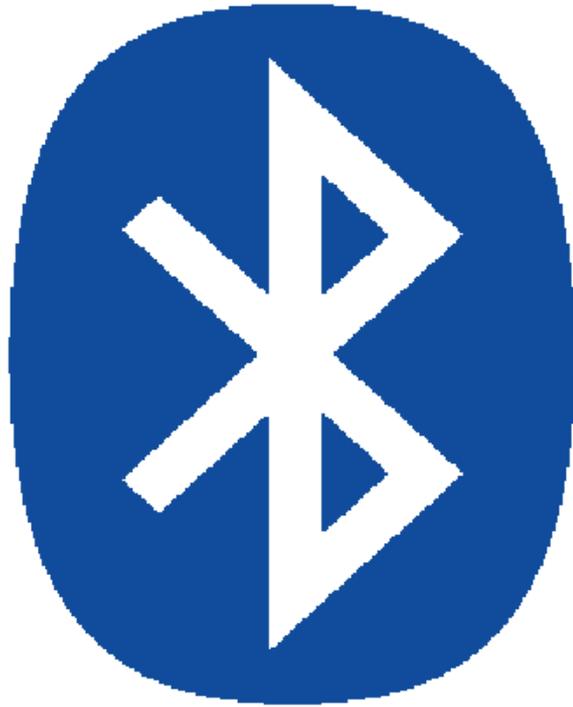


BLUETOOTH



le 11 juillet 2001

proposé

par

Philippe TRBICH

Table des matières

1	Introduction	4
2	Champ d'Application	5
2.1	Applications potentielles	5
2.2	Panorama des Produits Existants	6
2.2.1	Produits Commerciaux	6
2.2.2	Circuits Intégrés	7
2.2.3	Analyseurs de Protocole	7
2.2.4	Kits de Développement	7
2.2.5	Logiciels de Développement	8
2.3	Autres standards	8
3	Couche Bas Niveau	10
3.1	Couche matérielle	10
3.1.1	Support Radio	10
3.1.2	Contrôle de Puissance	12
3.1.3	Liens Physiques	12
3.2	Format de Paquet	13
3.2.1	Identification	13
3.2.2	Les Codes d'Accès	13
3.2.3	En-Tête	14
3.2.4	Corps, Charge	15
4	Couche Intermédiaire	16
4.1	Protocoles	16
4.1.1	Canaux Logiques	16
4.1.2	Protocole de Gestion de Liaison	16
4.1.3	Protocole L2CA	17
4.1.4	Routines de Transmission et de Réception	17
4.1.5	Audio	18
4.1.6	Synchronisation	18
4.1.7	Scatternet	18
4.2	Contrôle de Canal	19
4.2.1	Spécificité du Maître	19
4.2.2	Spécificité des Etats	19
5	Sécurité	21
5.1	Correction d'erreur	21
5.2	Equilibrage des paquets	22
5.3	Confidentialité	22
6	Couche Applicative	24
6.1	Protocole de Découverte de Service	24
6.2	Interfaces	25
6.2.1	RFCOMM	25
6.2.2	Téléphone	25
6.2.3	Interface de Contrôle d'Hôte	25
6.3	Extensions	27
7	conclusion	28

Chapitre 1

Introduction

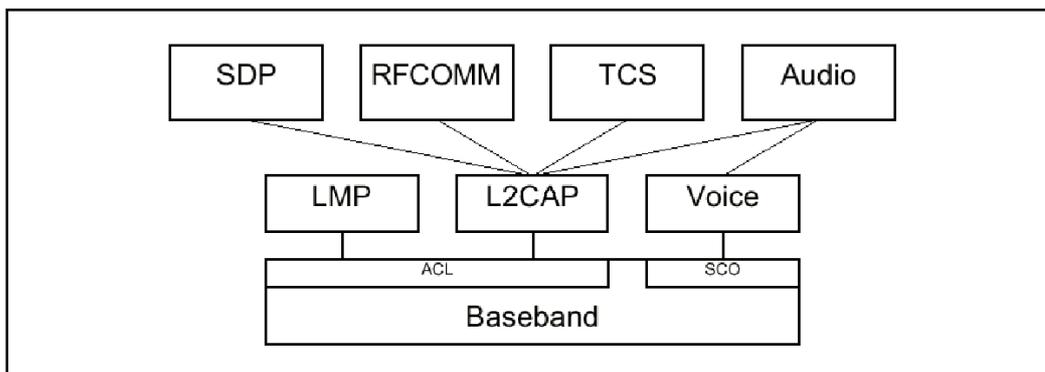
Bluetooth signifie littéralement “Dent Bleue” et provient de Harald Blaatand, chef viking, qui a été le fédérateur de différents clans au X^{ieme} siècle de la Norvège et du Dannemark. C’est la raison pour laquelle Ericsson, initiateur de Bluetooth, lui a donné ce nom en 1994.

C’est un protocole de transmission radio dont le domaine d’application est naturellement très étendu car il accompagne la mobilité de plus en plus grande des systèmes d’informations et simplifie leur communication : son usage est tant domestique que professionnel.

On le voit, les potentiels applications sont multiples et vont se développer car Bluetooth a été prévu dès le départ pour être à faible complexité, faible consommation, faible coût et grand champ d’applications.

De plus, le protocole est soutenu depuis 1998, dans le cadre d’un SIG (Special Interest Group), par les grands noms de la haute technologie : Intel, Nokia, IBM, Toshiba au début pour passer à plus de 2400 membres actuellement car Ericsson en a rendu les spécifications librement disponibles pour ceux qui s’engagent à les respecter.

Nous en développons ci-après quelque uns des éléments les plus caractéristiques, en commençant par l’offre déjà disponible dans ce domaine avec les champs d’applications déjà réalisés et ceux possibles de Bluetooth, puis nous commencerons à détailler le protocole lui-même en suivant le graphe ci-dessous, avec sa couche physique (baseband) et de protocole bas niveau (LMP et L2CAP) pour terminer sur les interactions avec les autres protocoles de communication (SDP, RFCOM et TCS).



Mais commençons par l’état actuel de l’art.

Chapitre 2

Champ d'Application

Ce protocole libère tous les périphériques en remplaçant leurs cables par la communication radio s'ils étaient déjà reliés et connecte ceux qui ne l'étaient pas.

L'intérêt d'un tel réseau est multiple :

- * Ericsson en a partagé les droits
- * le réseau accepte jusqu'à 8 membres et est facilement extensible à 80.
- * la connexion des nouveaux membres du réseau se fait automatiquement.
- * le débit maximum théorique peut aller jusqu'à 1 Mbits/s.
- * la partie purement électronique est très simple car un des objectifs est qu'il soit faible coût mais malgré cela, le système est résistant aux bruits.

De plus, le potentiel en volume d'unités est énorme car il comprend tous les produits ayant une partie électronique.

Ne serais-ce que pour le téléphone portable, une estimation faite par Ericsson en janvier 2001, table sur une intégration de 100 million de puce Bluetooth pour le premier semestre 2002, ce qui corrige les estimations de Dataquest pour ce même secteur de la téléphonie mobile, en date du 09 novembre 2000, données dans le tableau ci-dessous :

Année	2000	2001	2002	2003	2004
En millions d'unités	1.2	28.8	123.7	318	572
En pourcentage	0.3	5.3	18.5	40.8	65.4

Cette tendance est confirmée car on voit déjà que le nombre de produits homologués sur le site du SIG [4] est en forte accélération ces derniers mois. En effet, alors que la première homologation a eu lieu le 19/06/2000, pour avoir 240 produits au 21/06/2001, 74 des ces homologations ont eu lieu ce dernier mois.

2.1 Applications potentielles

Le volume potentiel est énormes car Bluetooth peut "tout" connecter. Si l'on pense seulement au domaine informatique, il permet de s'affranchir de tous les cables, que ce soit celui de la souris, du clavier, de l'imprimante, du modem : il est possible d'installer un système informatique dans une entreprise ou dans une maison sans avoir à disposer de cables et ce, d'une manière instantannée car les reconnaissances de périphériques se font automatiquement.

Un exemple tout simple mais montrant sa souplesse et sa puissance est le cas du modem, dont les débits actuels de transmission sur internet sont de 56kbits/s sur RTC (réseau téléphonique commuté), de 64kbits/s pour Numéris et 500kbits/s pour l'ADSL (Advanced Digital Subscriber Line). Il est possible d'installer un modem dans une pièce, proche de la ligne extérieure puis avoir un ordinateur dans n'importe quelle pièce, voire d'avoir un portable en déplacement, Bluetooth le gardera relié au modem.

Mais ce n'est qu'une des applications car sa souplesse est telle qu'il peut relier des appareils qui ne l'étaient pas jusqu'ici : le chauffage central, le téléphone portable avec l'autoradio de la voiture, la chaîne HIFI, la commande de la lumière...

Bluetooth peut relier un ensemble écouteur-micro avec le logiciel de reconnaissance vocale d'un ordinateur ou d'un portable qui permettra de téléphoner avec les mains libres soit en passant par ce dernier, soit en passant par le RTC s'il est à portée.

De la même manière, il permettra de d'augmenter la température de la pièce si la chaudière est reliée et le réfrigérateur invitera à valider une commande de courses s'il est vide.

Lors des déplacements en train, il reliera automatiquement le portable à la borne interne du TGV et lors de déplacements en avion, il le coupera automatiquement lors du décollage. Arrivé à destination, il permettra de réserver une chambre et la régler dans l'hôtel avec le portable ou le PDA (Personal Digital Assistant).

Cela devient réalité avec les produits proposés par les membres du SIG.

2.2 Panorama des Produits Existants

2.2.1 Produits Commerciaux

Depuis le lancement du SIG Bluetooth en 1998, 240 produits ont déjà obtenus la qualification de compatibilité répartis en trois champs d'applications :

* l'informatique :

- de nombreux fabricants proposent des cartes au format PCI comme Motorola [5] ou Toshiba [6].
- Inventel [10] et Elsa [7] ont dévoilé des modems permettant de connecter 7 périphériques.
- AXIS [8] propose un concentrateur d'impression qui accepte les liaisons réseau standard ainsi que les liaisons Bluetooth en cherchant les imprimantes sans fil dans son voisinage lors de sa mise sous tension. Il peut par exemple se connecter avec l'imprimante Deskjet 995C de HP [9] qui embarque Bluetooth.

* le professionnel :

- Toshiba propose un kit pour ses rétro-projecteurs et les affranchir des cables.
- Ericsson vend un "headset" [11], ensemble micro et écouteur ultra-léger qui peut se connecter soit à un portable (mais avec des puissances émises au niveau de l'oreille 1000 fois moindres), soit à un PC.
- Un hotel-restaurant Hollyday Inn sur Wall Street (New York-USA) permettra cet été d'enregistrer et de régler sa chambre d'hôtel ainsi que le restaurant avec un téléphone portable relié avec Bluetooth. Le portable servira aussi de clé pour la chambre. [14]

* les applications dans le domaine privé :

- IBM Japon [14] a développé un PDA prototype le 7/03/2001 sous forme d'une montre qui allie Bluetooth à Linux avec un écran de 640*480.
- TDK et Tactel [15] ont développé un module, nommé Blue5, qui se clippe sur un PDA de la série PalmV ou PalmVx, lui ouvrant le champ des applications Bluetooth.
- Essilor, en collaboration avec la société américaine The MicroOptical Corporation, a développé des lunettes informatives [12] qui, en intégrant un projecteur vidéo dans l'une des branches de la monture, permettent de transmettre une image à l'oeil. La liaison se fait par radio.

- Elsa, en collaboration avec Ford, a dévoilé au public un exemple de liaison entre un PC et une voiture électrique qui permet à cette dernière de régler les sièges en fonction de son occupant et de télécharger des fichiers mp3.

Et ce ne sont que quelques exemples de produits mais qui sont significatifs.

2.2.2 Circuits Intégrés

Avant tout, Bluetooth est d'abord un jeu de circuits intégrés et presque tous les fondeurs développent des solutions matérielles complètes ou partielles tant le marché des circuits Bluetooth semble prometteur.

En effet, le prix des composants devrait baisser de 25 USD actuellement à moins de 5 USD l'unité d'ici le premier semestre 2002 par effet de volume. Ce coût de 5 USD est symbolique car il correspond à la moitié du prix du cordon qu'il va remplacer. Et la maîtrise de la technique devrait permettre d'en accroître encore la fiabilité.

Il existe actuellement deux axes de développement :

- * les circuits qui implémentent Bluetooth d'une manière complètement matérielle

- La société ALPS [16] fournit un module complet pour la gestion transparente de la partie radio et logique de Bluetooth. Les entrées/sorties se font au choix par un port série ou USB et ce, pour un encombrement réduit à 32*15*3.5mm.

- Sharp [17] propose un système qui gère le protocole. Ici le circuit gère simplement un port série et une sortie pour un CODEC indépendant.

- C'est aussi le cas de la société Transilica [18] qui propose un circuit : ici, le circuit intègre toute la partie radio ainsi que la circuiterie logique sur la base d'un cœur de 8051 dans un circuit BGA (Ball Grid Area) 81 broches. Cette solution permet d'interfacer le circuit avec tout système électronique et informatique mobile en l'intégrant comme circuit spécifique.

- * les circuits qui utilisent des circuits radio spécifiques mais dont la logique est implémentée sur des bases plus standard.

Par exemple, Broadcom [19] ne fournit qu'un circuit radio sous forme d'un circuit 48 broches PLCC, aux spécifications respectant la norme mais nécessitant la circuiterie logique complémentaire.

2.2.3 Analyseurs de Protocole

Il est aussi nécessaire de pouvoir tester physiquement la qualité des liaisons radio, c'est pourquoi la société Anritsu [20] propose un jeu d'instruments de test et d'analyse qui fournissent une référence radio aux différentes puissances, couplé à un banc d'analyse qui détaille les couches du protocole.

2.2.4 Kits de Développement

Philips [21] commercialise un kit de développement qui comprend une carte mère sur laquelle viennent se connecter des cartes filles tant pour les circuits Base-band que pour les diverses solutions de circuits radio. Il est monitoré par un logiciel qui permet de tester les différentes caractéristiques des produits et de la qualité des liaisons.

Une autre société, IAR [22] propose aussi un kit sous forme d'une plateforme d'évaluation et de développement composé d'une carte mère portant un module radio Bluetooth et relié à l'extérieur par un connecteur de casque, un port série et un port USB, et par un logiciel qui permet soit de lancer des applications de démonstration, soit d'en développer de nouvelles.

2.2.5 Logiciels de Développement

Si Linux permet une connexion Bluetooth depuis son noyau 2.2, par la liaison USB, Windows a annoncé qu'il ne le proposerait pas sur son nouveau système d'exploitation XP car il préfère actuellement soutenir d'autres standards orientés spécifiquement sur l'informatique.

Néanmoins, l'offre de logiciels couplé à des circuits électroniques pour accroître l'intérêt de Bluetooth touche beaucoup de domaines mais deux ressortent :

Emulateurs de protocoles

Par exemple, la société SiliconWave [23] propose un logiciel d'interface qui émule la couche de gestion de protocole avec ses fonctionnalités tout en permettant des liaisons avec des API ou les couches applicatives telles que RFCOMM ou USB Transport tel qu'il l'est montré sur le schéma.

Il est en outre possible d'analyser les signaux pour le débogage en temps réel.

Un autre exemple est la société ADAMYA [24] qui, avec son logiciel propose une couche logicielle qui permet d'émuler toutes les spécifications Bluetooth (hors couche de gestion de protocole) ainsi que toutes les interfaces vers l'extérieur. Il est en plus multi-plateformes et est facilement portable puisqu'il est écrit en C.

Logiciels applicatifs

La plus part des logiciels applicatifs sont fournis avec les modules Bluetooth matériels vendus par les fabricants et sont spécifiques mais ils peuvent aussi utiliser des suites développées par des sociétés extérieures comme la société Digianswers [25] qui propose déjà un jeu de logiciels applicatifs et de liaison sous différentes versions de windows vers des cartes ou des composants Bluetooth.

Motorola et Toshiba les utilisent pour leur cartes PCI.

2.3 Autres standards

Dans le même domaine d'application, il existe d'autres standards pour faire des liaisons sans fil avec chacuns leurs avantages et inconvénients :

- * IrDA (Infrared Data Association) : ce système est basé sur la lumière infrarouge. Il possède l'avantage d'avoir un taux de transmission de 16 Mbits/s au maximum mais par contre, son champ d'application est restreint car son rayon d'action n'est que de 1.2 m (20 cm pour le débit de 16 Mb/s) et il doit être direct car l'infrarouge ne traverse pas les corps solides.
- * DECT (Digital Enhanced Cordless Telecommunication) : norme radio pour les téléphones domestiques, elle utilise la bande dédiée des 1.88 à 1.9 GHz en 10 bandes avec un schéma TDD. 24 slots de 10ms sont définis, la moitié pour les liaisons montantes et l'autre pour les liaisons descendantes. Cela permet de fournir 12 liaisons vocales full duplex (liaison simultanée dans les deux sens) (3 pour Bluetooth) à 32kbits/s par codage ADPCM.
Il est aussi possible de transférer des données à un taux maximum de 552 kbit/s en utilisant tous les canaux. Le système est constitué d'une base et d'un ou plusieurs mobiles. Lors d'une connexion, la base "ouvre" deux canaux et envoie les données sur les deux canaux. Celui qui a meilleure qualité est gardé. L'autre avantage du DECT est sa portée de 300 m.
- * IEEE 802.11a et IEEE 802.11b, deux standards radio issus du 802.11 qui sert de définition aux (réseau locaux sans fils). Ils utilisent la bande 2.4 GHz à 1 ou 2 Mbits/s avec une extension vers 5.5 et 11 Mbits/s et, comme Bluetooth, possèdent 79 canaux sur lesquels ils se déplacent à 50 sauts/s (1600 pour Bluetooth) pour une puissance d'émission pouvant aller jusqu'à 1 W.

Il existe actuellement une tentative de constitution d'un réseau MAN (Metropolitan Area Network) à Seattle mais s'il utilise les mêmes fréquences que Bluetooth, son domaine d'application, son mode de développement et la nécessité d'émetteurs fortement directionnels ne lui font pas concurrence et pourrait même étendre ses possibilités là où il est le plus faible, à savoir les hauts débits et les grandes distances.

- * HomeRF, sous forme d'un groupe de travail, pour spécifier un protocole de communications sans fils numériques entre des PC et des produits grands publics numériques. Le groupe est scindé en sous groupes pour s'adapter aux spécificités des normes locales (Japon notamment). Ce standard s'appuie sur les normes IEEE 802.11 et DECT en fonction du type de données à transporter, avec un taux de transfert actuel de 2 Mbits/s.

C'est le concurrent direct de Bluetooth car il possède la performance du DECT pour la transmission de voix et celle du 802.11 pour les réseaux sans fils locaux.

- * IEEE 802.15, standard défini juste après l'apparition de Bluetooth, pour les réseaux sans fils essentiellement à usage domestique. Il se repose sur la norme Bluetooth mais en l'ajustant aux standard IEEE.
- * HIPERLAN (Hight PERFORMANCE Local Area Network), est développé par l'ETSI (European Telecommunication Standards Institute). Il utilise la bande des 5 GHz et est constitué des Types 1 et Type 2. Le premier est plutôt un LAN Ethernet sans fil à 18 Mbits/s alors que le Type 2 est un ATM sans fil pour des débits allant jusqu'à 50 Mbits/s.

Les deux derniers standards sont donnés à titre indicatifs car encore en cours de développement mais ils sont révélateurs de l'importance que sont amenés à prendre les communications sans fils.

Après ce panorama succinct de l'état du développement des offres pour Bluetooth, voyons-en les spécifications.

Chapitre 3

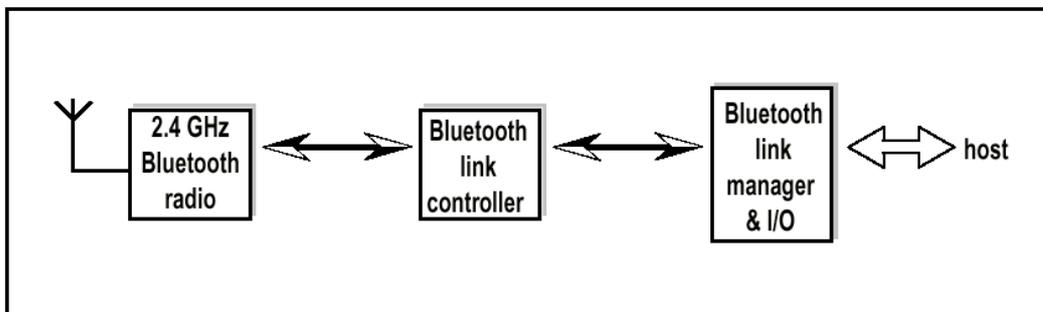
Couche Bas Niveau

3.1 Couche matérielle

En effet, non seulement Bluetooth possède une couche radio sur la bande des 2,4 GHz ISM (Industrial, Scientific and Medical) à très faible portée (jusqu'à 10 m de base), pouvant porter jusqu'à 100 m après amplification. Mais il est aussi constitué d'une couche logicielle qui, si elle ne respecte par la norme OSI (Open System Interconnect) à 7 couches, elle s'en rapproche. Cela permet de constituer un petit réseau, appelé aussi Piconet ou Scatternet dans le cas où un esclave est relié à plusieurs réseaux.

Bluetooth est composé de trois couches principales :

- une couche physique qui gère toutes les liaisons bas niveau
- une couche de contrôle de liaison qui définit tous les éléments du protocole qui lient les unités Bluetooth
- une couche de gestion qui relie Bluetooth au monde extérieur



3.1.1 Support Radio

Bluetooth utilise la bande des 2.4 GHz, avec quelques particularités en fonction de certains pays, comme indiqué dans le tableau ci-dessous.

Pays	Champ de Fréquence	Canaux RF	Nombre
Europe et USA	2400-2483.5 MHz	$f=2402 + k$ MHz	$k=0, \dots, 78$
France	2446.5-2483.5 MHz	$f=2454 + k$ MHz	$k=0, \dots, 22$

Pour limiter les pollutions sur les autres bandes, il existe une bande de garde basse de 2 MHz et une haute de 3,5 MHz, sauf pour la France où les bandes de garde sont de 7.5 MHz de chaque coté.

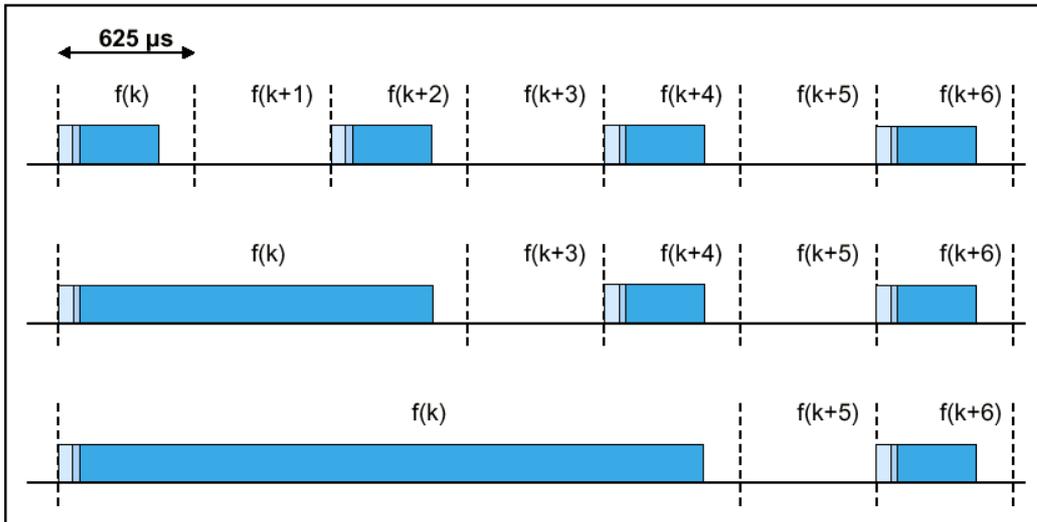
On peut voir que les possibilité de canaux RF sont limitées en France (23 canaux au lieu de 79). En effet, les fréquences supplémentaires sont occupées par l'armée mais des négociations sont en cours pour augmenter la bande utilisable.

Le nombre de canaux RF est important car le système Bluetooth repose sur une séquence de série de sauts parmi ceux-ci. Cette série de sauts ou canal est spécifique à un réseau piconet car il dépend directement de l'identification du maître. Chaque membre du piconet est synchronisé en temps et en saut.

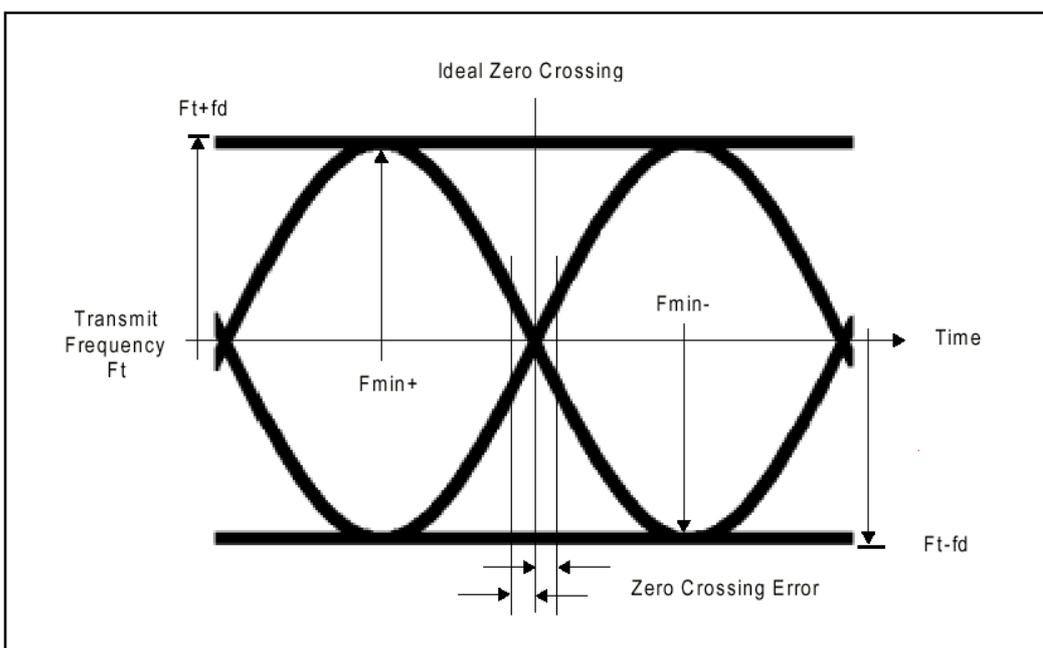
Le réseau ainsi constitué fait 1600 sauts à la seconde. Chaque saut consiste dans la sélection d'un canal RF d'un pas de 1 MHz dans lequel les données sont transmises dans un slot de temps. Cette séquence de sauts couvre tous les slots, ce qui permet une répartition uniforme de la puissance d'émission sur toute la bande. Chaque slot est référencé par rapport au maître du piconet. Les slots sont identifiés de 0 à $2^{27}-1$ et durent $625\mu s$. La séquence de sauts est cyclique avec une longueur de 2^{27} .

Le maître ne démarre la transmission que pendant les slots de temps pairs alors que les esclaves utilisent les slots impairs.

Le paquet de démarrage doit être avec le slot de départ mais la transmission montante ou descendante peut se faire sur une longueur pouvant aller jusqu'à 5 slots d'affilée, en gardant le même canal RF en fonction du type de paquets envoyés comme le montre le schéma.



Les données sont transmises à 1 Mbits/s sous forme d'une modulation binaire FSK avec un produit BT de 0.5. Le 1 binaire est représenté par une déviation positive et un 0 par une déviation négative de la fréquence. La déviation maximale doit rester entre 140kHz et 175kHz et ne doit en aucun cas descendre en-dessous de 115 kHz.



Ici l'erreur de passage par zéro doit être inférieure à $1/8^{ième}$ de symbole.

3.1.2 Contrôle de Puissance

Les puissances sont prises sur le connecteur d'antenne à cause des difficultés de mesure. Les puissances d'émissions supérieures à 0dBm doivent pouvoir être contrôlées (logiciellement) par le protocole de gestion des liens afin de limiter au minimum les puissances émises.

De plus, les puissances d'émission sont rangées en trois classes :

Classe	Puiss Maximum	Puiss Nominale	Puiss Minimum	Contrôle de Puiss
1	100 mW (20 dBm)	N/A	1 mW (0 dBm)	<+4 dBm à Pmax
2	2.5 mW (4 dBm)	1 mW (0 dBm)	0.25 mW (-6 dBm)	
3	1 mW (0 dBm)	N/A	N/A	

Le contrôle est optionnel de Pmin à Pmax pour les trois classes.

La sensibilité de la réception réelle est mesurée avec un niveau d'entrée pour lequel le taux d'erreur est en dessous de 0.1

Ce contrôle de puissance est une caractéristique essentielle de Bluetooth car elle permet, dans des espaces très faibles, d'avoir plusieurs réseaux indépendants sans trop d'inter-pollution et elle permet de limiter les consommations à 8 μ A en veille (mode Park) et de 8 à 30mA en émission (modes Sniff ou Hold). Ces caractéristiques sont celles définies de base mais les circuits, en fonction de leurs caractéristiques et de la qualité de la transmission, peuvent descendre nettement au-dessous.

3.1.3 Liens Physiques

Deux types de liens peuvent être initialisés: les liens SCO (Synchronous Connection-Oriented) et ACL (Asynchronous Connection Less).

SCO est un lien symétrique point à point établi par le maître en envoyant un message d'initialisation à un esclave spécifique où les slots sont réservés par le protocole LM . Ce message d'initialisation contient les données comme l'intervalle de slot Tsc0 et l'offset Dsc0. Il est principalement destiné aux informations limitées dans le temps comme la voix. C'est pourquoi les paquets ne sont jamais retransmis Le maître et les esclaves peuvent réserver jusqu'à trois liens sauf si ces derniers sont connectés à deux maîtres auquel cas le nombre de liens est limité à deux. Les paquets sont envoyés à intervalles réguliers Tsc0 aux esclaves dans des liens SCO réservés.

Cela permet d'avoir trois liaisons travaillant à 64 kbit/s.

ACL est un lien avec une durée de 1 slot. Il contient l'adresse d'identification de l'esclave. Dans le cas où un seul esclave est concerné, s'il réussit à décoder son adresse, il peut renvoyer des données vers le maître (la plus part du temps, il renvoie le message qu'il a reçu, ce qui permet de s'assurer de la validité des données). S'il n'y a pas d'adresse, le lien est considéré comme broadcast , c'est à dire que tous les esclaves sont concernés mais ils ne peuvent pas répondre.

Il permet des liaisons dissymétriques à 723.2 kbit/s d'un coté alors que l'autre ne sera qu'à 57.6 kbit/s ou symétriques à 433.9 kbit/s des deux cotés.

3.2 Format de Paquet

3.2.1 Identification

Chaque unité Bluetooth possède une adresse spécifique sur 48 bits, séparée en trois champs :

- LAP : partie basse de l'adresse sur 24 bits
- UAP : partie haute de l'adresse sur 8 bits
- NAP : partie de l'adresse non significative sur 16 bits.

Cette adresse est unique pour chaque composant et elle est utilisée dans les communications sous forme de paquets dont voici le format :

- * une zone pour le code d'accès de 72 bits
- * une zone d'en-tête de 54 bits
- * la zone de donnée ou charge de taille variable de 0 à 2745 bits max

Ce sont ici les tailles maximales et les bits les premier reçus sont de poids faible (LSB) .

LSB	72	54	0-2745	MSB
	Code D'accès	En-Tete	Charge	

3.2.2 Les Codes d'Accès

Ces codes d'accès permettent d'identifier tous les paquets échangés qui ont le même code de canal. Un corrélateur passe sur le code et déclenche un signal lorsqu'il a terminé. Ce signal permet de synchroniser le timing de réception.

Trois types de code d'accès ont été définis :

- IAC (Inquiry Acces Code) : Il existe sous deux formes, la première, GIAC pour "General" qui permet aux différents éléments de connaître tous les systèmes Bluetooth qui sont dans leur entourage proche et la seconde, DIAC pour "Dedicated" où la recherche des éléments est identique à celle du GIAC hormis que la recherche ne se fait que pour les éléments partageant les mêmes caractéristiques (il existe 63 possibilités de caractéristiques différentes).

- CAC (Channel Acces Code) : il permet d'identifier un piconet et est donc inclu dans tous les paquets de communication entre le maître et le ou les esclaves.

- DAC (Device Acces Code) : il n'est utilisé que dans les procédures spécifiques de signalement comme le paging ou la réponse au paging.

Ces codes d'accès sont constitués à partir de l'adresse basse du composant (LAP). Le CAC est suivi d'un en-tête et fait 72 bits de long. Les deux autres sans en-tête n'ont que 68 bits. Ils possèdent plusieurs rôles : ils permettent la synchronisation, la compensation de l'offset et bien sûr l'identification du piconet.

Le code d'accès est aussi utilisé dans les procédures de pagination et de recherche. Dans ce cas, le code d'accès est utilisé sans en-tête ni charge.

Il est lui-même constitué d'un préambule, d'un mot de synchronisation et d'une traine éventuelle.

LSB	4	64	64	MSB
	Préambule	Mot de Synchronisation	Traine	

Le préambule peut avoir deux motifs de 4 bits prévus pour la compensation comme ce qui suit dans le schéma : si le mot de synchronisation commence par un 1, le motif se terminera par un 0 et inversement.

LSB	MSB	LSB
1 0 1 0	1-----	

Préambule Mot de Synchro

LSB	MSB	LSB
0 1 0 1	0-----	

Préambule Mot de Synchro

La synchronisation est un mot de 64 bits, constitué de 24 bits d'adresse.

La traîne, optionnelle, est présente dès qu'il y a un en-tête. De même que le préambule, il est composé d'une suite de 0 et 1, en fonction du dernier bit du mot de synchronisation.

3.2.3 En-Tête

L'en-tête contient les informations sur le lien et possède 6 champs comme le montre le schéma ci-dessous.

LSB	3	4	1	1	1	8	MSB
AM_ADDR		TYPE	FLOW	ARQN	SEQN	HEC	

- * AM_ADDR : adresse temporaire de membre actif sur 3 bits. cette adresse est utilisée dans le sens montant et descendant. Si cette adresse est à 0, le maître fait du broadcasting. Les membres qui deviennent inactifs perdent leur adresse et se voient réaffecter une nouvelle adresse lorsqu'ils redevennent actifs.
- * TYPE : code de type sur 4 bits. 16 types de paquets peuvent être interprétés, après avoir déterminé si le lien est de type SCO ou ACL. Il indique le nombre de slots qui seront occupés pendant la transmission. Cela permet aux autres esclaves de connaître le temps pendant lequel ils n'auront pas à écouter.
- * FLOW : contrôle de flux ACL sur 1 bit : lorsque le buffer d'entrée est plein, le bit est positionné à 0 et renvoyé pour stopper temporairement la transmission des paquets ACL exclusivement.
- * ARQN : indicateur de réception sur 1 bit : lorsque le CRC (Cyclic Redundancy Check) indique une réception correcte de la charge, il est positionné à 1 (le 0 est positionné par défaut).
- * SEQN : numéro de séquence sur 1 bit
- * HEC : (header error check) contrôle d'erreur de l'en-tête sur 8 bits (voir la partie sécurité).

Sur les 2*16 types possibles, seuls 15 paquets sont définis, donc certains sont spécifiques aux modes ACL ou SCO : Les quatre premiers sont généraux et les autres sont détaillés par type de lien.

- * ID, qui ne contient que le DAC ou l'IAC pour les routines de requêtes et réponses
- * NULL, qui ne contient que le CAC pour transmettre les bits de contrôle
- * POLL, identique à NULL mais demande un acquittement
- * FHS, paquet de contrôle spécial qui informe sur l'adresse du composant Bluetooth et l'horloge de l'émetteur parmi d'autres informations. Ces informations sont dans la charge et protégées par un CRC 16 bits et codées avec un taux de 2/3 FEC. Il est utilisé pour la synchronisation de la fréquence des sauts lors du démarrage d'un nouveau piconet ou lors du changement de piconet. Voici ses différents champs :

34	24	2	2	2	8	16	24	3	26	3
Parity bits	LAP	Un-defined	SR	SP	UAP	NAP	Class of device	AM_ADDR	CLK ₂₇₋₂	Page scan mode

où

- les bits de parité forment le mot de synchronisation du code d'accès de l'unité envoyant le paquet FHS;
- LAP est la partie basse de l'adresse;
- SR indique la répétition du "scan";
- SP est le temps pendant lequel la réponse va être écoutée;

- UAP est la partie haute de l'adresse;
- NAP la partie non significative;
- Class of Device définit la classe du composant;
- AM_ADDR est l'adresse de membre actif;
- CLK₂₇₋₂ contient la valeur de l'horloge du début de la transmission du paquet FHS;
- le mode page scan indique le type de scan qui est fait par l'émetteur.

* DM1 (Data Medium rate), il peut être utilisé dans les deux types de transmission. Dans le cas de l'ACL, il ne transporte que des données utilisateurs (alors qu'il peut aussi transporter des données de contrôle pour les liens SCO) avec un codage 2/3 FEC et ajout de zéro pour que la charge reste à un multiple de 10 tout en restant sur 1 slot.

Les types de paquets ACL contiennent tous un CRC 16 bits (sauf si spécifié autrement) :

- * DH1, identique à DM1 sur un slot mais sans codage FEC (d'où le nom de Data Hight rate)
- * DM3, qui est un paquet DM1 dont la charge est étendue à 3 slots sur le même canal RF
- * DH3, identique au DM3 mais sans FEC
- * DM5, identique au DM1 avec une utilisation de 5 slots
- * DH5, identique au DM5 mais sans FEC
- * AUX1, identique au DH1 mais sans code CRC

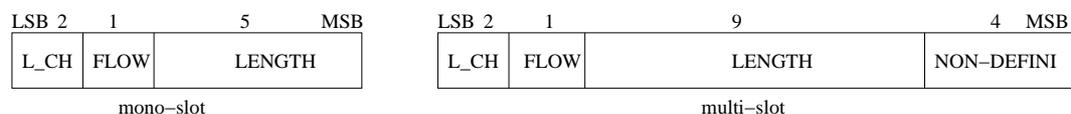
Et ceux pour SCO (sans CRC) ne sont jamais retransmis car leur champ d'application est la voix :

- * HV1 (Hight quality Voice), les octets sont protégés par un FEC 1/3 et chaque paquet porte 1.25ms de parole à 64kbts/s. Il faut donc envoyer 1 paquet pour 2 slots
- * HV2, les octets sont protégés par un FEC 2/3 et chaque paquet porte 2.5ms de parole à 64kbts/s. Il faut donc envoyer 1 paquet pour 4 slots
- * HV3, pas de protection par FEC et chaque paquet porte 3.75ms de parole à 64kbts/s. Il faut donc envoyer 1 paquet pour 6 slots
- * DV (Data Voice), mélange de voix sans FEC et de donnée avec un FEC 2/3. Des zéros doivent être rajoutés pour que la charge soit un multiple de 10

3.2.4 Corps, Charge

Dans la section précédente, nous avons vu que les données étaient surtout transmises par lien ACL et que la voix l'était par SCO (hormis les paquets DV qui ont les deux).

Dans le cas de la transmission de données, la charge est constituée d'un en-tête de charge (2 bits), d'un bit de FLOW, de 5 bits de longueur (en mono slot) ou de 9 bits avec 4 bits de traine (non encore définis) dans le cas du multislot, comme indiqué dans le schéma ci-dessous.



Le champ L_CH permet de valider à quelle couche de Bluetooth se fait l'envoi des données et comment. Ce point est développé dans le chapitre Protocoles.

L_CH	Canal Logique	Utilisation
00	Indéterminé	Sans
01	L2CAP	envoi d'un paquet fragmenté
10	L2CAP	envoi du premier paquet
11	LMP	message LMP

Le bit FLOW sert à contrôler le flux de données de la même manière qu'il l'a été vu plus haut. La longueur est celle des octets dans la charge sans l'en-tête et le CRC. Ensuite, il y a le corps des données significatives suivies par le CRC 16 bits.

Chapitre 4

Couche Intermédiaire

4.1 Protocoles

Il est à noter que lorsque nous parlons de maître et d'esclave, il ne s'agit que d'un protocole de communication car chaque "noeud" Bluetooth est identique et ce n'est que le protocole qui les différencie.

4.1.1 Canaux Logiques

Dans le système Bluetooth, 5 canaux logiques sont définis, dont trois utilisateurs (UA, UI et US) et deux de contrôle (LC et LM). Le canal LC est le seul à être porté par le paquet d'en-tête, les autres sont portés par l'en-tête du corps. En voici leur définition :

- * Canal LC (Link Control) : ce canal porte des informations sur le contrôle de la liaison bas niveau comme ARQ, le contrôle de flux et les caractères de la charge.
- * Canal LM (Link Manager) : il porte des informations de contrôle sur la gestion de la liaison entre le maître et l'esclave. Ce canal est indiqué avec L_CH mis à 11.
- * Canaux UA/UI (User Asynchronous/Isochronous data) : ils portent les données utilisateurs asynchrones en un ou plusieurs paquets fragmentés (dans le cas de multiples paquets, L_CH est mis à 10 dans le premier puis à 01 dans les suivants). Dans les canaux isochrones, les paquets sont envoyés à intervalles définis, réguliers (leur L_CH est aussi à 10).
- * Canal US (User Synchronous data) : Ce canal est porté par les liaisons SCO.

Après avoir abordé les couches basses des spécifications Bluetooth, avec la partie radio et le contrôle de liaison, la gestion de la liaison est assurée par la couche au-dessus, le protocole LM (Link Manager Protocol). Ce dernier ne communique qu'avec les couches les plus basses au contraire de son pendant, le protocole L2CA (Logical Link Control and Adaptation Protocol) qui sert de couche intermédiaire avec les protocoles de communication déjà existants pour transférer des données utilisateur.

4.1.2 Protocole de Gestion de Liaison

Comme il l'a été vu dans le chapitre sur les formats de paquets, les données destinées au LMP sont incluses dans la charge et sont prioritaires sur les données utilisateurs (L2CAP). Elles portent sur la gestion du piconet. Par exemple la gestion de l'établissement et destruction des liaisons, leur configuration, ainsi que les considérations de sécurité sont faites par ce protocole.

Il gère les modes Hold, Sniff et Park, ainsi que le contrôle de puissance.

Toutes les informations passées dans le corps du paquet se font sous la forme de PDU (Protocol Data Units) qui forment un jeu de messages sous forme de transactions. Ces messages occupent les 7 premiers bits du premier octet, le 8^{ième} indique le sens : 0 si l'initiateur du message est le maître et 1 pour l'esclave.

Le jeu de messages est proposé dans l'annexe F. Il est donné à titre indicatif car les noms des commandes permettent de comprendre leur objet.

4.1.3 Protocole L2CA

Ce protocole sert de couche de liaison entre la couche bas niveau et les protocoles extérieurs pour les liaisons de type ACL exclusivement (hors paquets AUX1). C'est pourquoi il fournit un certain nombre de services d'interface comme :

- * Le Multiplexage de Protocole: Avec les multiples possibilités de communication, comme le SDP (Service Discovery Protocol), RFCOMM et TC (Telephony Control), L2CA adapte les spécificités de chacun d'eux au réseau Bluetooth et inversement.
- * La Segmentation et le Ré-assemblage: Les paquets définis par Bluetooth peuvent être trop petits pour les protocoles extérieurs, L2CA se charge de segmenter les données et les ré-assembler dès leur réception.
- * Qualité de Service: L2CA doit s'assurer d'un minimum de qualité de service entre les unités Bluetooth par une bonne gestion des ressources.
- * Groupe: L2CA permet la gestion des groupes de piconets en redirigeant les données vers le bon réseau.

4.1.4 Routines de Transmission et de Réception

La transmission est faite de manière séparée pour les liaisons ACL et SCO par le biais de buffers. Si les esclaves ne possèdent que 2 buffers au plus (un par type de liaison), le master possède autant de buffers ACL que d'esclaves et un ou plusieurs buffers par esclave SCO.

Chaque buffer consiste en deux registres FIFO (First In, First Out) : un registre "courant" qui peut être lu et écrit par le contrôleur Bluetooth pour composer les paquets et un registre dit "suivant" qui est accessible par le Gestionnaire de Liaison pour charger de nouvelles informations. La commutation des registres est réalisée par le Contrôleur de liaison.

Dans le cas de transmission ACL, seuls les types de paquets DM ou DH sont utilisés. Pour terminer de valider une dernière donnée, un paquet NULL peut être envoyé. Dans les autres cas, aucun paquet n'est envoyé. Ici, seul le buffer ACL est utilisé. Le Gestionnaire de Liaison charge les nouvelles données dans le FIFO pointé par l'aiguillage puis il envoie une commande de traitement au Contrôleur de Liaison qui commute de FIFO. Lorsque la charge doit être envoyée, un paquet est composé en fonction du TYPE avec un en-tête et le corps y est adapté puis le tout est transmis. Lors de la réception de l'acquittement, il est vérifié si la transmission doit être réitérée. Si aucune donnée n'est dans la FIFO, un paquet NULL est envoyé.

Le premier traitement d'envoi est déclenché par une commande "flush". Si les FIFO sont régulièrement chargés, le traitement se fait automatiquement. Il n'est nécessaire de relancer une commande "flush" qu'en cas d'interruption de chargement des FIFO ou lors de passage en mode isochrone.

Pour la transmission SCO, le FIFO "suivant" du port synchrone est chargé en permanence pendant que "courant" est vidé cadencé à T_{sco} , établi lors du démarrage de la liaison SCO. Si un paquet de contrôle ou une liaison entre le maître et un autre esclave doit être faite, les données sont abandonnées pour celles de contrôle au format DM1.

Pour l'émission de paquets DV (mix de voix et de données), chaque buffer est chargé séparément. Ensuite le Contrôleur de Liaison charge les données puis la voix dans le paquet (la voix n'est retransmise qu'une fois mais les données peuvent être les mêmes en cas d'échec de réception). S'il n'y a pas de données à envoyer, le paquet est changé automatiquement en HV.

La réception est aussi gérée séparément pour les liaisons ACL et SCO. Mais contrairement à la transmission, il n'y a qu'un buffer pour tous les esclaves. Chaque buffer est constitué de deux registres FIFO, un qui est accessible par le Contrôleur de Liaison pour y charger des données, l'autre par le Gestionnaire de Liaison pour lire les données précédemment chargées. Ce processus est valide pour les deux types de liaisons ACL et SCO. Le contrôle de flux se fait par le bit de FLOW.

4.1.5 Audio

Bluetooth peut supporter jusqu'à trois canaux audio à 64 kbits/s simultanément. Les canaux, spécifiques aux liaisons SCO, peuvent avoir plusieurs formats en fonction de la négociation entre les Gestionnaires de Liaison :

- PCM Log (Pulse Coded Modulation) avec une compression loi A ou loi μ . La transformation d'une loi vers l'autre se réalise aisément pour respecter les formats des différents canaux.
- CVSD (Continuous Variable Slope Delta Modulation) avec compensation syllabique.

4.1.6 Synchronisation

Les échanges de données, synchronisées sur l'horloge du maître, démarrent toujours sur les slots pairs pour le maître (CLK1=0) et sur les slots impairs pour les esclaves.

La dérive de temps pour slot de 625 μ s, ne doit pas dépasser 20 ppm et son jitter doit être inférieur à 1 μ s. Ce sont les esclaves qui adaptent leur horloge avec un offset, mis à jour à chaque réception de paquet, pour correspondre avec l'horloge du maître. Cet offset ne doit pas dépasser 10 μ s sauf dans le cas où l'esclave était en mode "hold" (mode dans lequel l'esclave n'envoie, ni de reçoit de message) auquel cas, il peut utiliser une partie de sa fenêtre de transmission et de réception pour se synchroniser et émettre au slot suivant.

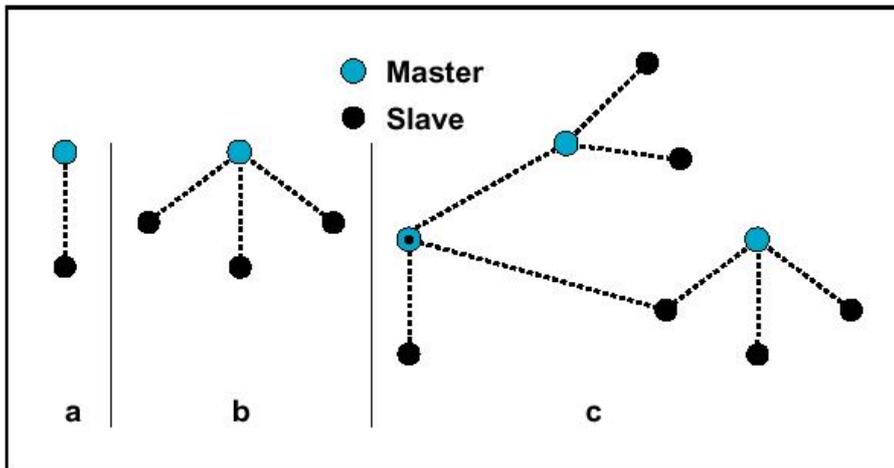
Le mode park possède un fonctionnement similaire lors du réveil de l'esclave.

4.1.7 Scatternet

Plusieurs piconets peuvent couvrir la même zone géographique avec des maîtres, des séquences de saut, etc, différents.

Ce sont les maîtres qui différencient les piconets les uns des autres avec leurs identifications spécifiques; mais malgré cela, il peut y avoir des collisions puisque la bande de fréquence est la même pour tous.

Un esclave ou un maître peuvent participer à deux piconets, c'est ce que l'on appelle un scatternet à partir du moment où il reste au moins esclave dans l'un des piconets. La communication entre deux piconets peut être facilitée en commutant un esclave et un maître dans un piconet dans le cas où c'était le maître qui assurait la liaison entre les deux piconets : la procédure est définie comme un piconet switch. En voici l'illustration :



Dans les liaisons, ACL, les esclaves peuvent se placer en mode hold, sniff ou park dans un des piconets pendant qu'il devient actif dans l'autre.

S'il y a une liaison SCO, cela ne peut se faire qu'avec des paquets HV3, 1 slot sur 4 car un des slot est réservé pour la synchronisation à cause des décalages des deux horloges des maîtres et deux autres pour le changement de piconet.

4.2 Contrôle de Canal

Pour qu'un réseau piconet se constitue, il faut au moins qu'il y ait connexion entre un maître et un esclave, répondant à un protocole défini.

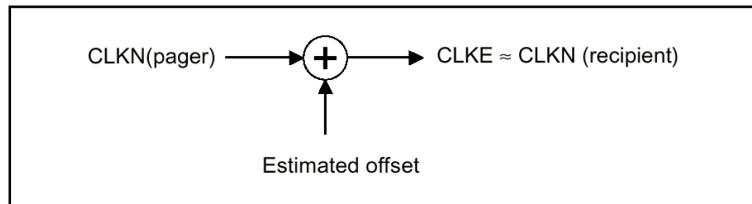
4.2.1 Spécificité du Maître

Le piconet est entièrement défini par le maître car c'est son adresse qui indique la séquence de sauts de fréquence et le code d'accès de canal. C'est lui qui démarre la séquence de connexion vers un ou plusieurs esclaves.

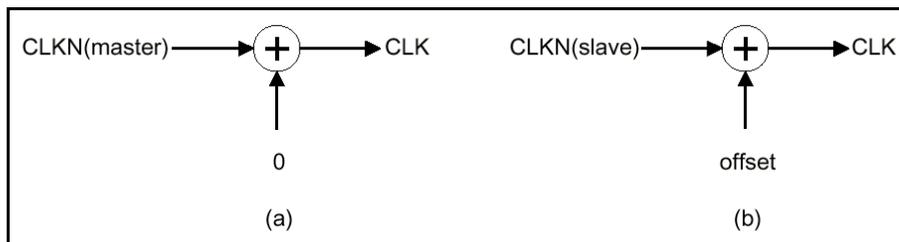
En effet, c'est son horloge qui synchronise tous les transferts dans le piconet. C'est pourquoi elle n'est jamais ajustée ni arrêtée. Les seuls changements pouvant avoir lieu sur l'horloge sont les offsets pour se synchroniser avec les esclaves. Mais cette horloge peut être de différents types :

- CLKN sert de référence à toutes les autres horloges et sa dérive doit être inférieure à 20 ppm. Dans les modes standby, hold, park, l'horloge peut être fournie par un oscillateur faible puissance avec une précision de +/- 250ppm.

- CLKE est l'horloge estimée à laquelle on ajoute un offset pour la synchronisation avec les esclaves.



- CLK est l'horloge du maître du piconet, dérivé de CLKN par l'ajout d'un offset qui doit être mis à jour régulièrement. C'est l'horloge qui est commune à tous les éléments du piconet.



4.2.2 Spécificité des Etats

Il existe deux états majeurs : Standby avec sept sous-états page, page scan, inquiry, inquiry scan, master response, slave response, inquiry response qui permettent l'addition ou le réveil des esclaves et Connexion avec les modes active, sniff, hold, park :

Standby

C'est l'état par défaut.

- * page scan : cet état correspond à l'ouverture d'une fenêtre d'écoute de son propre DAC pour 32 sauts de fréquence définis. Chaque écoute d'une fréquence se fait pendant 1.28 s maximum avec un intervalle de temps de repos de 2.56 s entre chaque scan.
- * page : il est utilisé par le maître pour se connecter avec un esclave qui est en mode page scan en envoyant le DAC de l'esclave régulièrement sur différentes fréquences prédéfinies. Leurs horloges ne sont pas synchronisées et dès que l'esclave est contacté, le nombre de sauts est porté à 3200/s car pour chaque saut, de 312.5µs, le maître a le temps d'envoyer un paquet d'ID (68 bits) et permettre une synchronisation plus rapide et plus fiable des horloges.
- * inquiry : C'est la procédure utilisée pour découvrir quels sont les nouveaux éléments Bluetooth à portée de radio. Le message continuellement envoyé à toutes les unités potentielles indique quelle classe d'unité peut répondre (tous si un GIAC est envoyé et seulement quelques uns si un DIAC est envoyé). Cette procédure est interrompue lors des liaisons SCO.

- * **inquiry scan** : C'est l'état dans lequel se met une unité pour être découverte si elle le souhaite. La procédure est à peu près identique à celle de page scan hormis le fait que c'est le GIAC qui est recherché.
- * **slave response** : Après avoir reçu son propre DAC, il le renvoie au maître par le même canal fréquentiel. Le maître et l'esclave sont synchronisés. Sur la séquence de saut suivante, l'esclave attend le paquet FHS du maître qu'il valide lors de la fenêtre de transmission suivante. La connexion est établie.
- * **master response** : Le maître rentre dans cette routine lorsque l'esclave a ré-émis son DAC. Il transmet le paquet FHS, indiquant l'horloge du maître, son adresse 48 bits et la classe de l'unité puis il attend la validation de réception de l'esclave. Si elle n'est pas reçue, le paquet FHS est renvoyé. Dès que la liaison est validée, le maître reprend BD_ADDR pour définir un nouveau canal de séquences de sauts et envoi comme premier paquet un POLL, le dialogue peut commencer.
- * **inquiry response** : dans cette opération de recherche, seuls les esclaves répondent. C'est pendant les périodes de réception que le maître obtient des messages de réponse avec un paquet FHS contenant tous les paramètres de l'esclave. Pour éviter les collisions lors de réponses simultanées, les esclaves calculent un nombre aléatoire qui leur donne le retard à appliquer en terme de nombre de sauts pour la réponse. Cette étape est recommencée jusqu'à ce que le maître ait validé l'envoi.

Connexion

Ici les liaisons sont déjà établies et les paquets peuvent être transmis ou reçus. Les deux unités utilisent l'horloge et le CAC du maître et la séquence de sauts définie. La connexion démarre par un paquet POLL de vérification et dure jusqu'à une commande detach (logiciel) ou un reset (matériel).

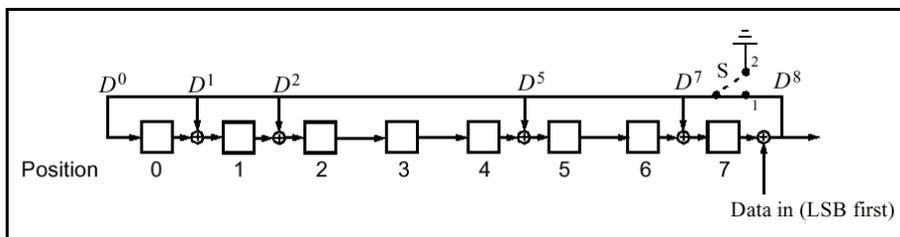
- * **active** : Dans ce mode, les unités participent activement à la transmission en fonction de la planification du maître. S'ils ne sont pas concernés par les messages, ils peuvent se mettre en mode d'attente mais doivent régulièrement synchroniser leur horloge en écoutant le trafic.
- * **sniff** : Ce mode permet de réduire l'activité d'écoute des esclaves. Par exemple, lors d'une transmission ACL, des slots sont dédiés à des esclaves spécifiques. C'est le maître qui envoie les informations avec le protocole LM (il envoie un offset D_{sniff} et une durée T_{sniff} pendant lequel l'esclave est concerné).
- * **hold** : C'est un mode pendant lequel l'esclave ne peut avoir de liaison ACL (mais les liaisons SCO peuvent être gardées). Après avoir informé le maître de la durée, cela permet à l'esclave de faire du scanning, du paging, de tenter de joindre un autre piconet. Pendant ce mode, l'esclave peut aussi se mettre en veille (faible consommation). Dans ce mode, il garde son adresse de membre actif AM_DDR.
- * **park** : Si l'esclave n'a pas besoin de rester actif, il peut se mettre dans ce mode de veille basse consommation où il perd son AM_ADDR mais obtient deux nouvelles adresses de parc : PM_ADDR qui est son adresse de membre dans le parc à l'intention du maître et AR_DDR dans le cas où la sortie du parc se fait à l'initiative de l'esclave. Les esclaves sortent régulièrement de ce mode pour se synchroniser et pour vérifier s'il n'y a pas de messages broadcast dans un slot qui leur est spécifique: le slot beacon. Il est à noter que cela permet d'augmenter le nombre de membres du piconet, tant qu'il n'y en a que 7 d'actifs au même moment, par la méthode du swapping.

Chapitre 5

Sécurité

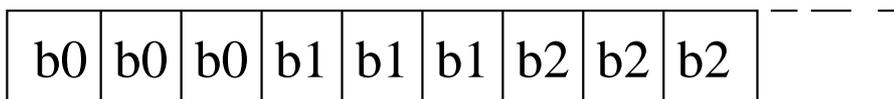
5.1 Correction d'erreur

Bluetooth pouvant transmettre dans des environnements bruités et pour limiter le nombre de retransmissions, il existe des possibilités de corrections d'erreurs de plusieurs types. Certaines sont systématiques, comme l'en-tête qui est systématiquement protégé par le HEC, les registres 0 à 8 prépositionnés avec les 8 bits de l'UAP :

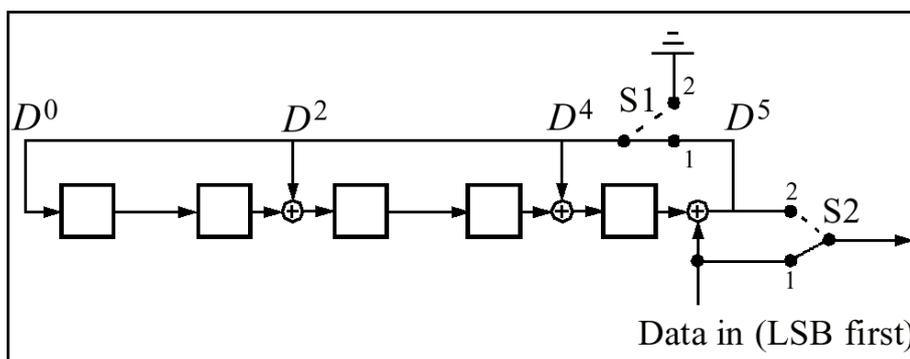


et par un :

- FEC 1/3 : en plus de l'utilisation systématique pour les en-têtes, il est aussi utilisé pour les paquets HV1. Le codage en lui-même consiste en une triple répétition du code :



- FEC 2/3 : c'est à base de code de Hamming (15, 10). Le polynôme générateur est $g(D) = (D+1)(D^4+D+1)$. Il peut corriger toutes les erreurs simples et détecter les erreurs doubles. Il est utilisé pour les paquets de données de DM, DV, FHS, HV2. En voici la technique de codage.



- ARQ: le sigle signifie Automatic ReQuest. L'information est retransmise tant qu'elle n'a pas été validée dans le paquet de retour (ARQN=1). Pour déterminer si les informations sont correctes, on ajoute un CRC aux données. Les en-têtes et la transmission de voix ne sont pas protégés par ARQ. La réponse de l'esclave se fait dans son slot, juste après la transmission du maître. En cas de succès, le bit SEQN est inversé alors que dans l'autre, le bit est resté à sa position initiale et le récepteur sait si la donnée doit être renvoyée.

La validation se fait après le contrôle du HEC et du CRC s'il est présent. Lors du démarrage d'une connexion, le maître envoie un POLL avec un ARQ à 0; l'esclave le renvoie de la même manière. Le bit ARQ n'est affecté que par les paquets contenant un CRC et des slots vides. Dans les paquets FHS, ainsi que dans la transmission broadcast, le bit ARQ n'est pas contrôlé.

Dans le cas où les données ne sont toujours pas récupérées au bout d'un certain nombre de retransmissions, le paquet est abandonné pour transmettre le suivant: le transfert de données est isochrone. Le récepteur est forcé pour prendre le paquet suivant et abandonner le précédent. L_CH (de l'en-tête de la charge) sera initialisé à 10.

Dans le cas de la transmission broadcast, l'adresse AM_ADDR de l'en-tête est initialisée à 0. Ici le processus de correction d'erreur est réalisé par la retransmission plusieurs fois du même paquet avant d'envoyer le suivant. Dans le cas où il existe un CRC, les paquets possèdent un numéro avec le bit SEQN (qui est initialisé à 1 au démarrage).

Comme il n'existe pas de validation pour les paquets broadcast, il est donc important de bien recevoir le premier. Le contrôle d'erreur se fait d'abord avec le HEC dans l'en-tête: le mot de synchronisation de 64 bits provient des 24 bits LAP (Low Address Part) du maître, on peut d'abord contrôler s'il n'y a pas d'erreur de piconet. Ensuite, HEC et CRC permettent à la fois de contrôler les erreurs et les mauvaises adresses: même si le contrôle du LAP est validé, un contrôle supplémentaire est fait sur l'UAP (Up Address Part).

5.2 Equilibrage des paquets

Avant toute transmission, l'en-tête et la charge sont mélangés pour limiter la composante continue et pour rendre aléatoire les motifs du message. Ce mélange est réalisé avant le codage FEC. Le mot de mélange est généré par le polynôme $g(D)=D^7+D^4+1$ et un XOR est effectué avec l'en-tête et la charge. Avant chaque transmission, le registre à décalage est initialisé avec l'horloge du maître CLK_{6-1} , étendu avec un 1 pour le MSB.

5.3 Confidentialité

La transmission se fait par les airs et nécessite donc un cryptage pour que les données restent dans le domaine privé. C'est pourquoi dans chaque unité Bluetooth sont implémentées les mêmes routines d'authentification et de cryptage. Quatre entités sont utilisées pour maintenir la confidentialité: une clé publique, deux clés secrètes et un nombre aléatoire.

- BD_ADDR, l'adresse du composant sur 48 bits qui est l'adresse publique. Elle peut être obtenue par une routine "inquiry".

- Une clé utilisateur privée pour l'authentification: de longueur 128 bits, elle possède une taille fixe. Elle est définie au démarrage du piconet puis n'est plus divulguée par la suite.

- Une clé utilisateur privée pour le cryptage: de longueur variable de 1 à 16 octets, pour respecter les législations en vigueur dans les différents pays ainsi que pour la mise à jour plus facile de la qualité de cryptage en fonction de l'évolution technologique. Elle est générée à partir de la clé d'authentification à chaque fois que le cryptage est activé mais tout en restant entièrement différente.

- Un nombre aléatoire de 128 bits: chaque élément Bluetooth possède son propre générateur aléatoire de nombres.

Les spécifications Bluetooth ne permettent pas de générer les nombres aléatoires de manière logicielle et doivent utiliser les générateur internes des circuit intégrés afin d'éviter que les utilisateurs substituent la génération aléatoire de chiffres par une version logicielle pseudo-aléatoire, donc de moindre efficacité.

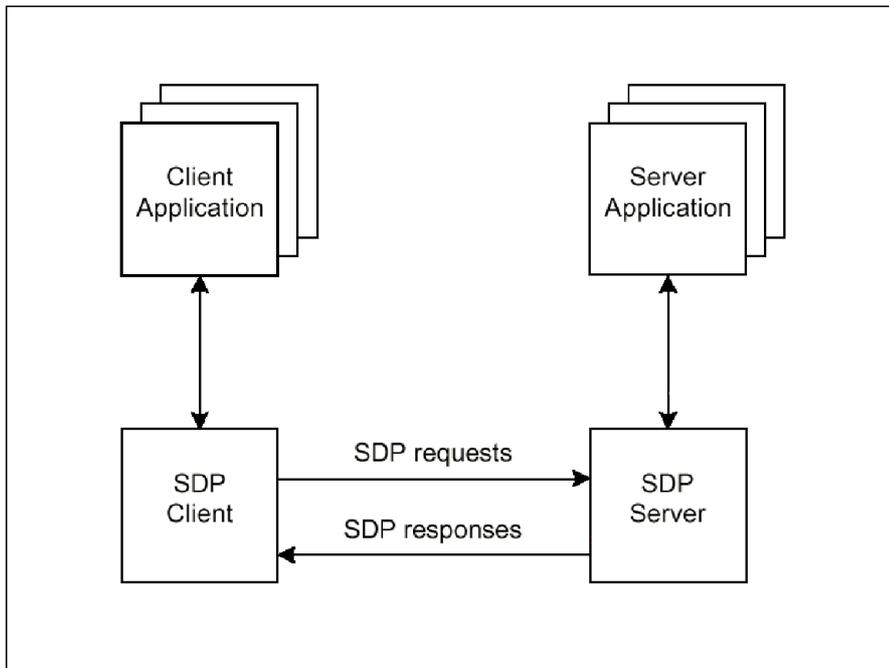
Chapitre 6

Couche Applicative

6.1 Protocole de Découverte de Service

Dans le cas d'un réseau filiaire normal, les connexions sont faites pour des durées longues, de quelques jours à quelques années. Dans le cas de Bluetooth, ces liaisons peuvent ne durer que quelques minutes.

Ces transactions sont faites sur le mode client-serveur, comme indiqué ci-dessous :



Le SDP permet aux applications de découvrir quels sont les services et leurs caractéristiques disponibles à portée du champ d'action de la radio pour les éléments mobiles.

Il doit permettre aux clients de chercher, de proposer et d'annuler des classes de services spécifiques en fonction de leur proximité radio et avec une identification spécifique. Toute cette gestion doit être faite automatiquement et c'est pourquoi elle constitue un protocole s'appuyant sur celui du L2CA.

Les étapes pour une connexion sont les suivantes :

- Recherche de connexion L2CAP vers un piconet par l'envoi de paquets sans charge
- Recherche d'une classe spécifique de services ou balayage des services proposés
- Réception des attributs nécessaires pour la connexion avec le service choisi
- Etablissement d'une connexion normale pour utiliser le service

Les serveurs SDP sont constitués de tout composant Bluetooth qui a des services à offrir et qui en maintient les caractéristiques dans une base de donnée dont chaque élément en est répertorié par des

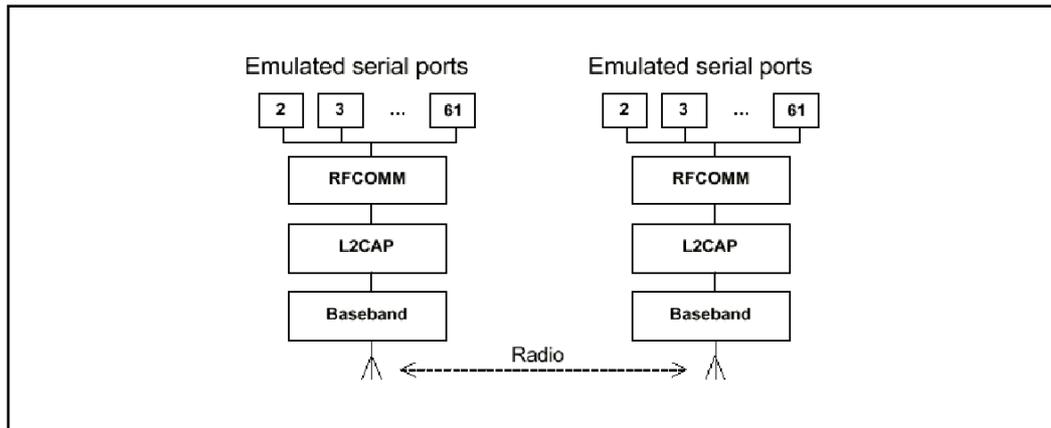
attributs. Chaque attribut est composé de deux champs, l'un 16 bits pour distinguer les services entre eux dans la table, nommé Attribute ID et l'autre, de taille variable est nommé Attribute Value, déterminé par l'Attribute ID et le type de service qui est proposé.

6.2 Interfaces

6.2.1 RFCOMM

Le protocole RFCOMM propose une émulation des ports séries vers L2CAP, en respectant le standard TS 07.10.

Il peut gérer jusqu'à 60 connexions simultanées entre deux composants Bluetooth en fonction de l'implémentation matérielle car il permet de remplacer les UART (Universal Asynchronous Receiver Transmitter) présents ou bien de simuler la liaison s'ils sont présents (par exemple, il peut simuler la liaison vers un modem série).



Il permet aussi, optionnellement, de gérer des liaisons vers de multiples éléments Bluetooth.

6.2.2 Téléphone

Bluetooth permet aussi le contrôle d'une liaison téléphonique via le TCS Binary (Telephony Control protocol Specification) avec les fonctionnalités suivantes :

- Contrôle d'Appel qui permet le contrôle de la synchronisation de la parole, c'est à dire les périodes de connexion et de déconnexion.
- Gestion de Groupes autorisant la connexion en mode multi-point (lors d'un appel, la connexion peut être dirigée vers plusieurs téléphones) ou en mode point-à-point (lors d'un décrochage, une connexion est établie entre le téléphone et la ligne téléphonique).
- TCS Sans Connexion qui gère tous les éléments hors connexion, à savoir la sonnerie vers tous les téléphones dans le champ radio, l'identification de l'appelant, l'arrêt de la sonnerie lors du décrochage, la transmission du numéro...

TCS n'est qu'une couche de protocole et il doit être piloté par une application servant d'interface utilisateur. Il est à noter qu'il est possible d'avoir plusieurs communications téléphoniques simultanées en démarrant plusieurs instances de TCS.

6.2.3 Interface de Contrôle d'Hôte

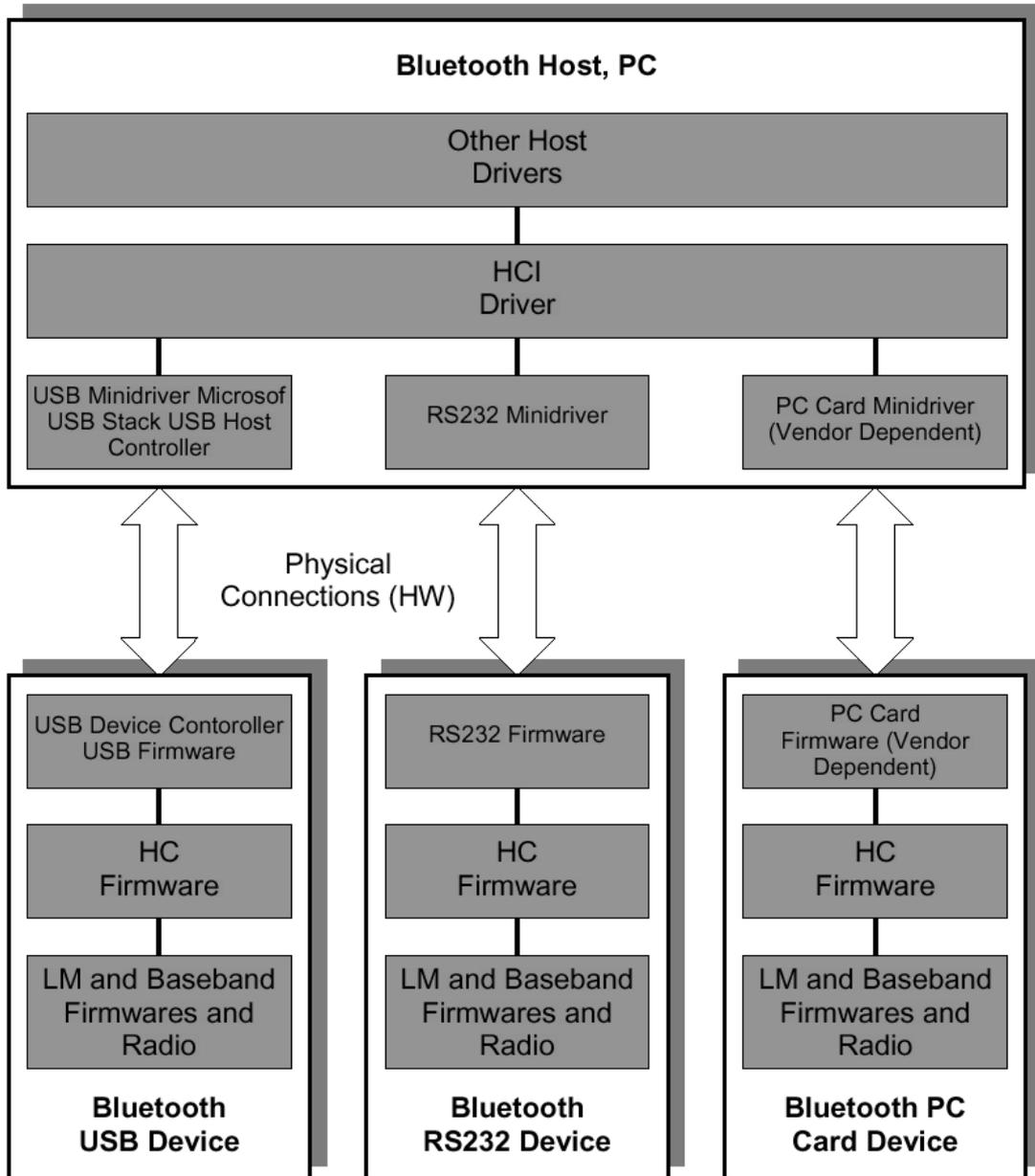
Il peut être souhaitable que la partie matérielle Bluetooth soit la plus simple possible pour des raisons :

- de facilité de développement (le circuit est standard donc le développement en est simplifié),
- d'intégration (la partie matérielle est réduite à sa plus simple expression donc prend moins de place),
- de coûts (la simplification permet la polyvalence et les coûts d'industrialisation sont moindres) ou
- pratiques (il peut lui être demandé de simplement "réveiller" l'hôte avec lequel il est relié et il n'a donc pas besoin d'être sophistiqué).

Pour ce faire, il est possible de n'avoir que la partie radio avec son Contrôleur et son Gestionnaire de Liaison sur une carte spécifique. Le reste du protocole peut être déporté sur un hôte (typiquement un PC) qui se charge de l'interface avec l'application, au prix d'une interface supplémentaire, nommée HCI (Host Controller Interface).

HCI se charge de la gestion de la connexion physique entre l'hôte et la carte Bluetooth. Ces liaisons peuvent être de plusieurs types : un port RS232, un port USB, un connecteur PCI...

Et comme il l'est spécifié sur le schéma ci-dessous, HCI doit être implanté des deux cotés de la liaison.



6.3 Extensions

Il a été fait mention de quelques interfaces de Bluetooth un peu plus haut mais ce n'est qu'un aperçu de ses possibilités. En effet, un de ses intérêts est qu'il laisse une grande ouverture aux autres standards avec le protocole OBEX (Object Exchange) qui permet l'échange simple et spontané de données. Ainsi, Bluetooth est déjà relié avec IrDA, protocole de transmission infrarouge et permet des liaisons WAP (Wireless Application Protocol) par le biais d'une liaison PPP (Point to Point Protocol).

Chapitre 7

conclusion

On le voit, Bluetooth a un avenir tout tracé dans le domaine de la communication sans fil.

Il n'en reste pas moins que le SIG travaille toujours à améliorer la norme et les caractéristiques du système: il est prévu une implémentation de la version 2.0 qui est en cours de développement.

Cette évolution permettra une meilleure fiabilité des connexions entre appareils Bluetooth et surtout, une sécurisation accrue dans la transmission.

L'objectif est de pouvoir utiliser le réseau pour faire du télépaiement à l'aide des terminaux qui voient le jour actuellement sur le marché (PDA, Smartphones...).

Une amélioration du débit est aussi en cours pour qu'il passe à 10Mbits/s afin d'étendre le champ d'application du picoréseau vers des applications de plus en plus multimédia.

Mais le plus important est que Bluetooth est une des composantes essentielles du mouvement qui nous propose de vraiment rentrer dans l'ère du tout numérique, en complément avec tous les autres standard.

A ceci près que Bluetooth le fera pour notre quotidien, tant dans notre environnement privé, que public ou professionnel. S'il est prévu 100 millions de puces pour la téléphonie mobile au premier semestre 2002, c'est des dizaines de milliard de puces qui forment le marché potentiel.

Après la "révolution" internet commence l'évolution Bluetooth.

En effet, si internet a permis de créer un "village global", c'est Bluetooth qui permettra aux "villageois" d'y avoir accès en toutes occasions.

Bibliographie

- [1] JENNIFER BRAY AND CHARLES F STURMAN. *Bluetooth: Connect Without Cables*. Éd. Prentice Hall
- [2] SIG BLUETOOTH. *Bluetooth 1.1 Specification Book*. Éd. SIG Bluetooth
- [3] SIG BLUETOOTH. *Bluetooth 1.1 profile Book*. Éd. SIG Bluetooth
- [4] [HTTP://WWW.BLUETOOTH.COM](http://www.bluetooth.com)
- [5] MOTOROLA *BTMPC100/BTMPC101 Bluetooth Mini Card*.
- [6] TOSHIBA *Bluetooth Card Kit*.
- [7] [HTTP://WWW.NETACTUEL.COM](http://www.netactuel.com) AUTEUR : JEAN-REN GONTHIER. *Bluetooth, un nouveau standard*.
- [8] AXIS *AXIS 5800 Mobile/Network Print Server*.
- [9] HP *Deskjet 995C*.
- [10] [HTTP://WWW.TRANSFERT.NET](http://www.transfert.net)
- [11] [HTTP://WWW3.WEBDO.COM](http://www3.webdo.com) AUTEUR : NICOLAS WILLEMIN. *Une oreillette sans fil/avril 2001*.
- [12] [HTTP://WWW.SNOF.ORG](http://www.snof.org) *luninformativ.html le 29/08/2000*
- [13] [HTTP://WWW.SPEKA.NET](http://www.speka.net).
- [14] [HTTP://WWW.WIRELESS.FR](http://www.wireless.fr).
- [15] [HTTP://WWW.TDKSYSTEMS.COM](http://www.tdksystems.com) AUTEUR : DAVE CURL *TDK Systems strengthens grip on handheld Bluetooth market le 11/06/2001*.
- [16] ALPS *Bluetooth Module Design Application*.
- [17] SHARP *Device Specification for Bluetooth Module Model DC2D1BZ*.
- [18] [HTTP://WWW.TRANSILICA.COM](http://www.transilica.com) *OneShipTM Technical Specification*.
- [19] [HTTP://WWW.BROADCOM.COM](http://www.broadcom.com) *BMC 2002, Product Brief*.
- [20] [HTTP://WWW.ANRITSU.COM](http://www.anritsu.com) *MT8850A Bluetooth Test Set*.
- [21] [HTTP://WWW.SEMICONDUCTORS.PHILIPS.COM](http://www.semiconductors.philips.com) *Blueberry Developers Kit for Bluetooth Applications*.
- [22] [HTTP://WWW.IAR.COM](http://www.iar.com) *Bluetooth Starter Kit*.
- [23] [HTTP://WWW.SILICONWAVE.COM](http://www.siliconwave.com) *Odysseytm Software*.
- [24] ADAMYA *C-Blue*.
- [25] [HTTP://WWW.DIGIANSWER.COM](http://www.digianswer.com) *Bluetooth Card Kit*.

Index

- active, 20
- ADSL, 5
- AM ADDR, 15
- AR ADDR, 20
- ARQ, 22
- Attribute ID, 25
- Attribute Value, 25
- AUX1, 15

- BD ADDR, 19, 22
- beacon, 20
- BGA, 7
- broadcast, 12

- CAC, 13
- Call Control, 25
- Canal LC, 16
- Canal LM, 16
- Canal US, 16
- canaux RF, 10
- Canaux UA/UI, 16
- CL TCS, 25
- Class of Device, 15
- CLK, 15, 19
- CLKE, 19
- CLKN, 19
- Connexion, 19
- CRC, 14
- CVSD, 18

- DAC, 13, 19
- DECT, 8
- DH1, 15
- DH3, 15
- DH5, 15
- DM1, 15
- DM3, 15
- DM5, 15
- DV, 15

- ETSI, 9

- FEC 1/3, 21
- FEC 2/3, 21
- FHS, 14
- FIFO, 17
- full duplex, 8

- Gestion de Groupes, 25
- GIAC, 13
- Groupe, 17

- HCI, 26
- HEC, 21

- HIPERLAN, 9
- hold, 18–20
- HomeRF, 9
- HV1, 15
- HV2, 15
- HV3, 15

- IAC, 13
- ID, 14
- IEEE 802.11, 8
- IEEE 802.15, 9
- inquiry, 19
- inquiry response, 20
- inquiry scan, 20
- IrDA, 8, 27
- ISM, 10
- isochrone, 22

- L CH, 15
- L2CAP, 15, 16
- LAN, 8
- LAP, 13, 14, 22
- LM, 12
- LMP, 15, 16
- LSB, 13

- MAN, 8
- master response, 20
- Multiplexage de Protocole, 17

- NAP, 13, 15
- NULL, 14

- OBEX, 27
- OSI, 10

- page, 19
- page scan, 19
- park, 19, 20
- PCM, 18
- PDA, 6
- PDU, 16
- Piconet, 10
- piconet, 18
- piconet switch, 18
- PM ADDR, 20
- point-to-point, 12
- POLL, 14
- PPP, 27

- Qualité de Service, 17

- Ré-assemblage, 17
- RFCOMM, 25

RTC, 5

Scatternet, 10

scatternet, 18

SDP, 17, 24

Segmentation, 17

slave response, 20

sniff, 20

SP, 14

SR, 14

Standby, 19

standby, 19

swapping, 20

TCS Binary, 25

Telephony Control, 17

time slot, 11

TS 07.10, 25

UAP, 13, 15, 21, 22

UART, 25

WAP, 27

WLAN, 8

Annexe A

Commandes du Protocole LM

LMP PDU	Length (bytes)	op code	Packet type	Possible direction	Contents	Position in payload
LMP_accepted	2	3	DM1/DV	m ↔ s	op code	2
LMP_au_rand	17	11	DM1	m ↔ s	random number	2-17
LMP_auto_rate	1	35	DM1/DV	m ↔ s	-	
LMP_clkoffset_req	1	5	DM1/DV	m → s	-	
LMP_clkoffset_res	3	6	DM1/DV	m ← s	clock offset	2-3
LMP_comb_key	17	9	DM1	m ↔ s	random number	2-17
LMP_decr_power_req	2	32	DM1/DV	m ↔ s	for future use	2
LMP_detach	2	7	DM1/DV	m ↔ s	reason	2
LMP_encryption_key_size_req	2	16	DM1/DV	m ↔ s	key size	2
LMP_encryption_mode_req	2	15	DM1/DV	m ↔ s	encryption mode	2
LMP_features_req	9	39	DM1/DV	m ↔ s	features	2-9
LMP_features_res	9	40	DM1/DV	m ↔ s	features	2-9
LMP_host_connection_req	1	51	DM1/DV	m ↔ s	-	
LMP_hold	7	20	DM1/DV	m ↔ s	hold time, hold instant	4-7
LMP_hold_req	7	21	DM1/DV	m ↔ s	hold time, hold instant	4-7
LMP_incr_power_req	2	31	DM1/DV	m ↔ s	for future use	2
LMP_in_rand	17	8	DM1	m ↔ s	random number	2-17
LMP_max_power	1	33	DM1/DV	m ↔ s	-	

LMP PDU	Length (bytes)	op code	Packet type	Possible direction	Contents	Position in payload
LMP_max_slot	2	45	DM1/DV	m ↔ s	max slots	2
LMP_max_slot_req	2	46	DM1/DV	m ↔ s	max slots	2
LMP_min_power	1	34	DM1/DV	m ↔ s	-	
LMP_modify_beacon	11 or 13	28	DM1	m → s	timing control flags	2
					D_B	3-4
					T_B	5-6
					N_B	7
					Δ_B	8
					D_{access}	9
					T_{access}	10
					$N_{acc-slots}$	11
					N_{poll}	12
					M_{access}	13:0-3
access scheme	13:4-7					
LMP_name_req	2	1	DM1/DV	m ↔ s	name offset	2
LMP_name_res	17	2	DM1	m ↔ s	name offset	2
					name length	3
					name fragment	4-17
LMP_not_accepted	3	4	DM1/DV	m ↔ s	op code	2
					reason	3
LMP_page_mode_req	3	53	DM1/DV	m ↔ s	paging scheme	2
					paging scheme settings	3
LMP_page_scan_mode_req	3	54	DM1/DV	m ↔ s	paging scheme	2
					paging scheme settings	3

LMP PDU	Length (bytes)	op code	Packet type	Possible direction	Contents	Position in payload
LMP_park_req	17	25	DM	m → s	timing control flags	2
					D _B	3-4
					T _B	5-6
					N _B	7
					Δ _B	8
					PM_ADDR	9
					AR_ADDR	10
					N _{Bsleep}	11
					D _{Bsleep}	12
					D _{access}	13
					T _{access}	14
					N _{acc-slots}	15
					N _{poll}	16
M _{access}	17:0-3					
access scheme	17:4-7					
LMP_preferred_rate	2	36	DM1/DV	m ↔ s	data rate	2
LMP_quality_of_service	4	41	DM1/DV	m → s	poll interval	2-3
					N _{BC}	4
LMP_quality_of_service_req	4	42	DM1/DV	m ↔ s	poll interval	2-3
					N _{BC}	4
LMP_remove_SCO_link_req	3	44	DM1/DV	m ↔ s	SCO handle	2
					reason	3
LMP_SCO_link_req	7	43	DM1/DV	m ↔ s	SCO handle	2
					timing control flags	3
					D _{sco}	4
					T _{sco}	5
					SCO packet	6
air mode	7					

LMP PDU	Length (bytes)	op code	Packet type	Possible direction	Contents	Position in payload
LMP_set_broadcast_scan_window	4 or 6	27	DM1	m → s	timing control flags	2
					D _B	3-4
					broadcast scan window	5-6
LMP_setup_complete	1	49	DM1	m ↔ s	-	
LMP_slot_offset	9	52	DM1/DV	m ↔ s	slot offset	2-3
					BD_ADDR	4-9
LMP_sniff_req	10	23	DM1	m ↔ s	timing control flags	2
					D _{sniff}	3-4
					T _{sniff}	5-6
					sniff attempt	7-8
					sniff timeout	9-10
LMP_sres	5	12	DM1/DV	m ↔ s	authentication response	2-5
LMP_start_encryption_req	17	17	DM1	m → s	random number	2-17
LMP_stop_encryption_req	1	18	DM1/DV	m → s	-	
LMP_supervision_timeout	3	55	DM1/DV	m → s	supervision timeout	2-3
LMP_switch_req	5	19	DM1/DV	m ↔ s	switch instant	2-5
LMP_temp_rand	17	13	DM1	m → s	random number	2-17
LMP_temp_key	17	14	DM1	m → s	key	2-17
LMP_timing_accuracy_req	1	47	DM1/DV	m ↔ s	-	
LMP_timing_accuracy_res	3	48	DM1/DV	m ↔ s	drift	2
					jitter	3
LMP_unit_key	17	10	DM1	m ↔ s	key	2-17

LMP PDU	Length (bytes)	op code	Packet type	Possible direction	Contents	Position in payload
LMP_unpark_BD_ADDR_req	variable	29	DM1	m → s	timing control flags	2
					D _B	3-4
					AM_ADDR 1 st unpark	5:0-2
					AM_ADDR 2 nd unpark	5:4-6
					BD_ADDR 1 st unpark	6-11
					BD_ADDR 2 nd unpark	12-17
LMP_unpark_PM_ADDR_req	variable	30	DM1	m → s	timing control flags	2
					D _B	3-4
					AM_ADDR 1 st unpark	5:0-3
					AM_ADDR 2 nd unpark	5:4-7
					PM_ADDR 1 st unpark	6
					PM_ADDR 2 nd unpark	7
LMP_unsniff_req	1	24	DM1/DV	m ↔ s	-	
LMP_use_semi_permanent_key	1	50	DM1/DV	m → s	-	
LMP_version_req	6	37	DM1/DV	m ↔ s	VersNr	2
					Compld	3-4
					SubVersNr	5-6
LMP_version_res	6	38	DM1/DV	m ↔ s	VersNr	2
					Compld	3-4
					SubVersNr	5-6