Enseignant responsable : Etienne DURIS

Michel VONGVILAY **Gabriel NGUYEN NGOC** **Grégory WOLOWIEC**

I. INTRODUCTION

Qu'est-ce que Bluetooth ? C'est une technologie de réseau personnel sans fil (noté WPAN pour Wireless Personnel Area Network), c'est-à-dire une technologie de réseaux sans fil à faible portée (quelques dizaines de mètres). Elle permet de relier plusieurs appareils entre eux sans liaison filaire, en utilisant les ondes radio comme support de transmission (bande de fréquence des 2,4 GHz).

Pour remonter aux origines de la technologie Bluetooth, il nous faut faire un lointain retour en arrière, aux années 60 : le port série. Le port série a été inventé afin de relier des périphériques (clavier, terminaux, matériels de mesure) à des ordinateurs : c'est le standard RS232.

Son évolution sans fil est l'IrDA, un protocole qui utilise les ondes lumineuses infrarouges pour la transmission de données. Cependant, tout comme le protocole RS232, il est possible de relier un seul périphérique à la fois. C'est pourquoi a été conçu l'USB (Universal Serial Bus) qui permet de relier plusieurs périphériques en série. Une évolution sans fil de ce protocole a naturellement vu le jour : Bluetooth. Cette technologie est donc une évolution lointaine sans fil du RS232.

Afin de comprendre Bluetooth, nous allons tout aborder quelques généralités, puis nous verrons la couche applicative de Bluetooth. Sa couche physique sera étudiée en troisième partie, ensuite la dernière partie sera consacrée à l'accès au médium. Nous conclurons enfin sur les limitations de Bluetooth.

II. GENERALITES

II.1. Bluetooth SIG

La technologie Bluetooth a été à l'origine mise au point par Ericsson en 1994. En 1998, un groupe d'intérêt baptisé Bluetooth SIG (Bluetooth Special Interest Group) a été fondé par Ericsson, IBM, Intel, Toshiba et Nokia. Aujourd'hui, plus de 2500 entreprises ont rejoint le groupe.

L'origine de l'appellation Bluetooth fait référence à un roi danois Harald « Dent bleue » (dû à son goût immodéré pour les mûres) qui aurait unifié les différents royaumes nordiques à la fin du Moyen Âge, de même que Bluetooth SIG s'est créé autour d'intérêts communs.

Le but principal du Bluetooth SIG est de développer des produits inter opérables. C'est ainsi qu'à été créée une spécification sans licence pour ses membres afin développer des produits et logiciels utilisation Bluetooth (standard IEE 802.15).

II.2. Description

Bluetooth a pour principal objectif de remplacer les câbles. En effet, les fils qui permettent de relier des périphériques à des ordinateurs par exemple sont assez souvent contraignants, et ne permettent pas une grande liberté de mouvement en plus d'être encombrants. C'est pour cela que cette technologie supporte les caractéristiques suivantes :

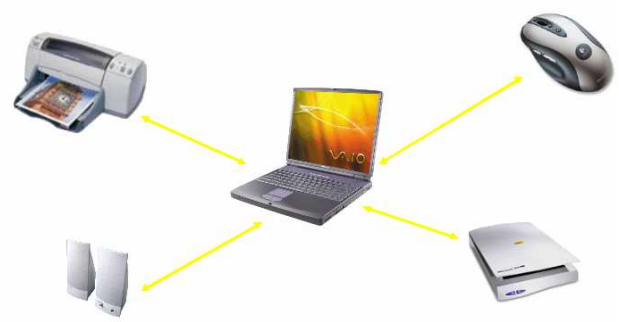
- Faible coût ;
- Faible puissance d'émission, d'où :
  - Courte distance d'émission (quelques dizaines de mètres)
  - Faible consommation d'énergie (donc adapté aux produits portables)
- Performances modestes (1Mbps) ;
- Topologie ad hoc ;
- Configurable dynamiquement ;
- Support des transferts voix et données ;
- Destiné à un usage personnel (PAN : Personal Area Network) ;
- Certification Bluetooth pour assurer la compatibilité des produits entre eux.

II.3. Usage

Bluetooth est aujourd'hui utilisé dans de nombreux secteurs dont voici quelques exemples :

- *Périphériques informatiques sans fil*

On peut désormais utiliser clavier, souris et casque audio en toute liberté.



- *Téléphonie mobile*

Il est possible de se connecter à partir de son ordinateur portable ou PDA vers un téléphone GSM avec fonction de modem et obtenir une connexion Internet.

Une autre fonction plus répandue est l'apparition des kits main libre Bluetooth (oreillette).

- *Synchronisation de périphérique*

La synchronisation des contacts et calendrier des PDA via Bluetooth est également possible.

- Automobile

L'application la plus parlante en automobile est l'apparition des récepteurs GPS Bluetooth. Associé à un Pocket PC et à un logiciel de navigation, cet ensemble permet d'équiper son véhicule d'un système de guidage GPS à moindre coût.

Il existe sur certaines voitures des kits main libre, permettant ainsi d'utiliser son téléphone mobile via l'écran multifonctions du véhicule en toute sécurité : accès au répertoire, numérotation…

- Domotique

La domotique permet de « piloter » une maison via un ordinateur. Il est par exemple possible de gérer le chauffage électrique, le lave-linge, le lave-vaisselle, la filtration d'une piscine, l'arrosage automatique… Bluetooth peut ainsi être utilisé dans ce domaine, afin de minimiser le câblage.

Bluetooth est donc une technologie permettant de supprimer les câbles entre périphériques. Afin que tous les périphériques estampillés Bluetooth puissent communiquer entre eux, la spécification Bluetooth 1.0 a été mise en place (qu'ils soient de marques différentes ou non). Nous allons maintenant nous intéresser plus particulièrement à la couche applicative de cette spécification, qui introduit le concept de « profils ».

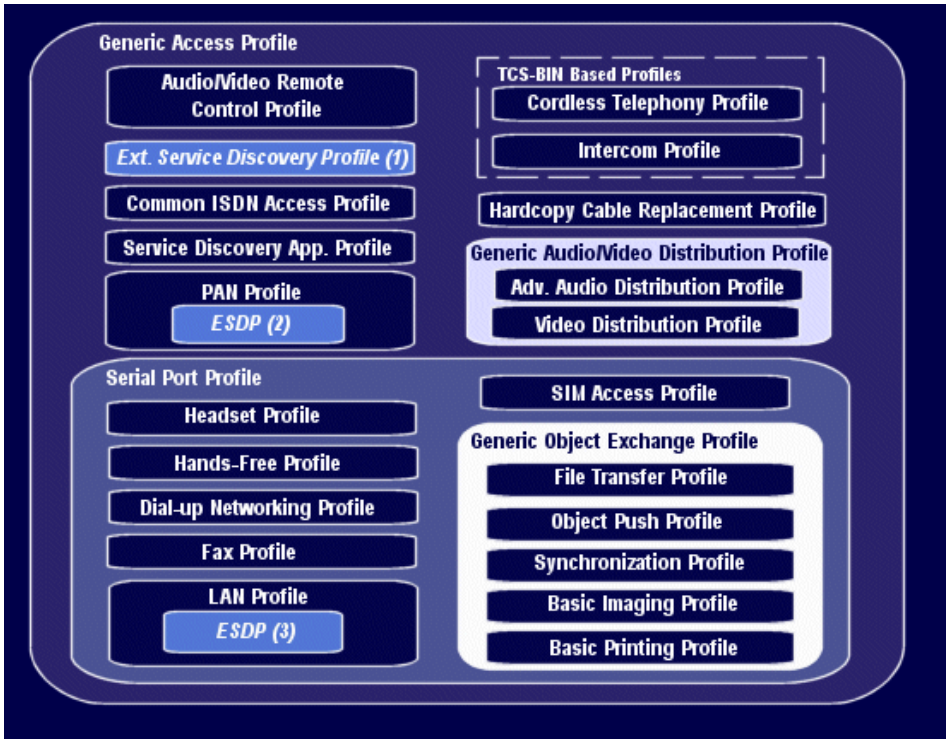
III. LA COUCHE APPLICATIVE : LES PROFILS

Le concept de profils est utilisé afin d'assurer le maximum de compatibilité entre les produits des différents constructeurs de produits Bluetooth. Ainsi, tous auront les mêmes modèles utilisateurs dans leur couche logicielle : on aura pour tous les appareils Bluetooth les mêmes appellations pour chaque fonctionnalité supportée.

Les profiles Bluetooth ont donc été développés afin de décrire comment implémenter les modèles utilisateur.

III.1. Hiérarchie des profils

Il existe une hiérarchie entre profil, et donc des dépendances entre eux. Pour illustrer ce phénomène, observons le schéma suivant :



Ainsi, le File Transfert Profil est dépendant du Generic Object Exchange Profile, du Serial Port Profile, et du Generic Access Profile.

III.2. Generic Access Profile

Ce profil est LE profil de base qui doit être implémenté par tous les appareils Bluetooth. En effet, c'est celui qui définit les procédures génériques de découverte d'équipement, ainsi que de gestion de connexion aux autres appareils Bluetooth.

Pour chaque profil, il existe plusieurs points qui sont redéfinis ou non : rôle, scénario, principes de base…

Rôles

On utilise les termes de :

- *Initiateur*
- *Accepteur*

Pour les 2 rôles que les protagonistes d'une communication Bluetooth peuvent prendre.

L'initiateur est celui qui pour une procédure donnée, est à l'origine de l'établissement d'un lien ou d'une transaction sur un lien existant.

Scénario

Un utilisateur Bluetooth doit en principe pouvoir se connecter à n'importe quel autre appareil Bluetooth, même si ils n'ont aucune application en commun. Cela doit être possible en utilisant les fonctions basiques de Bluetooth. En effet, il n'y a aucune application commune entre une oreillette Bluetooth Logitech et un téléphone mobile Nokia par exemple.

Principes de base de ce profil

Ce profil expose l'ensemble des caractéristiques de tous les équipements Bluetooth :

- Il expose les spécifications sur la représentation des propriétés Bluetooth : l'adresse Bluetooth, le nom d'un équipement, son type, le PIN number utilisé pour authentifier 2 périphériques ;
- Il définit les « modes » génériques à tous les profiles : discoverability mode (on peut le détecter), connectability mode (on peut s'y connecter), pairing mode (on peut créer un lien avec) ;
- Il définit les procédures générales qui peuvent être utilisées pour « découvrir » les propriétés basiques des équipements Bluetooth (nom, type…) qui sont « découvrables » ;
- Il décrit les procédures générales de connexions à d'autres dispositifs Bluetooth ;
- Il définit la procédure générale de création de liens entre des dispositifs Bluetooth ;

Ce profil est celui dont tous les autres dépendent, et tous les profils « héritent » de ses caractéristiques.

Nous allons maintenant passer en revue quelques autres profils importants.

III.3. Service Discovery Application Profile

Ce profil décrit les fonctionnalités et procédures d'une application ou périphérique Bluetooth afin qu'il puisse « découvrir » les services associés à d'autres périphériques Bluetooth et récupérer toute information relative à ces services.

Il définit également les protocoles et procédures à utiliser par une application de détection de services sur un périphérique pour localiser des services disponibles sur d'autres périphériques Bluetooth activés.

Rôles

Local device : c'est le périphérique qui instancie la procédure de détection de service. Un *local device* contient le Service Discovery Application, utilisé par l'utilisateur pour lancer la détection et afficher les résultats de la recherche.

Remote device : c'est un périphérique qui participe au processus de détection en répondant au *local device*. Un *remote device* contient une base de données des services qu'il propose afin de satisfaire à la requête du *local device*.

Ces rôles ne sont ni permanents ni exclusifs. Ces rôles sont fonction du périphérique qui est à l'origine de la transaction. Un périphérique peut donc à un instant donné jouer le rôle de *local device* en demandant les services disponibles par les autres périphériques, puis l'instant d'après être *remote device* en répondant à un autre périphérique en mode *local device*.

III.4. Serial Port Profile

Ce profil est un autre profil principal : en effet, c'est celui qui définit les protocoles et procédures qui doivent être utilisées par les périphériques utilisant Bluetooth pour émuler le protocole RS232 (connexion par câble série, ce que Bluetooth est appelé à remplacer).

De ce profil dépendent les suivants :

- Headset profile : utilisation des casques sans fil ;
- Dial up networking profile : permet d'utiliser un périphérique Bluetooth en tant que pont Internet (possibilité de se connecter à Internet à partir d'un Pocket PC via un téléphone GSM Bluetooth) ;
- Fax profile : envoi/réception de fax via un téléphone GSM Bluetooth
- Etc.…

III.5. Generic Object Exchange Profile

Ce profil définit les spécificités des modèles utilisateur d'échanges d'objets entre périphériques Bluetooth : carte de visite, synchronisation, transfert de fichier…

File Transfert Profile

Ce profil est utilisé par les applications de transfert de fichier (comme son nom l'indique). Un exemple typique est une connexion d'un Pocket PC vers un téléphone avec fonction d'appareil photo, puis récupération des photos du téléphone vers le Pocket PC.

Synchronisation Profile

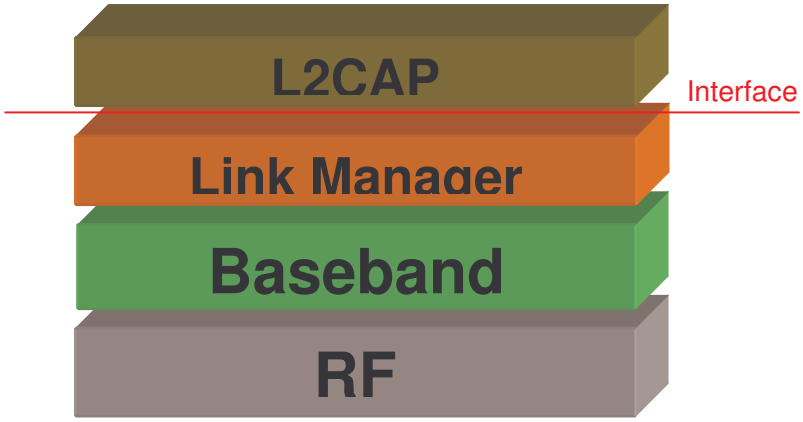
Ce profil va permettre à un PDA de synchroniser ses données avec une station de base via Bluetooth (comme il pourrait le faire via port série, USB ou IrDA).

Il existe d'autres profils Bluetooth permettant de définir d'autres modèles utilisateurs (utilisation d'un récepteur GPS par exemple), et d'assurer la compatibilité de tous les équipements implémentant ces profils entre eux.

Nous allons maintenant voir la couche physique de la technologie Bluetooth.

IV. PRESENTATION DE LA COUCHE PHYSIQUE

Comme présenté au dessus, la couche physique est constituée de cette façon :



Les éléments fondamentaux d'un produit Bluetooth sont définis dans les deux premières couches protocolaires, la couche radio et la couche bande de base. Ces couches prennent en charge les tâches matérielles comme le contrôle du saut de fréquence et la synchronisation des horloges.

IV.1. La couche radio fréquence RF

La couche radio (la couche la plus basse) s'occupe de l'émission et de la réception des ondes radio. Elle définit les caractéristiques telles que la bande de fréquence et l'arrangement des canaux, les caractéristiques du transmetteur, de la modulation, du receveur, etc.

La technologie Bluetooth utilise l'une des bandes de fréquences ISM (Industrial, Scientific & Medical) réservée pour l'industrie, la science et la médecine. La bande de fréquences utilisée est disponible au niveau mondial et s'étend sur 83,5 MHz (de 2,4 à 2,4835 GHz).

Cette bande est divisée en canaux de 1 Mhz soit 79 canaux au total. Pour transmettre des datas, la technologie Bluetooth utilise le FHSS (Frequency Hopping Spread Spectrum).

Le principe du FHSS est la commutation rapide entre plusieurs canaux de fréquence, utilisant un ordre pseudo aléatoire connu tant à l'émetteur qu'au récepteur pour la

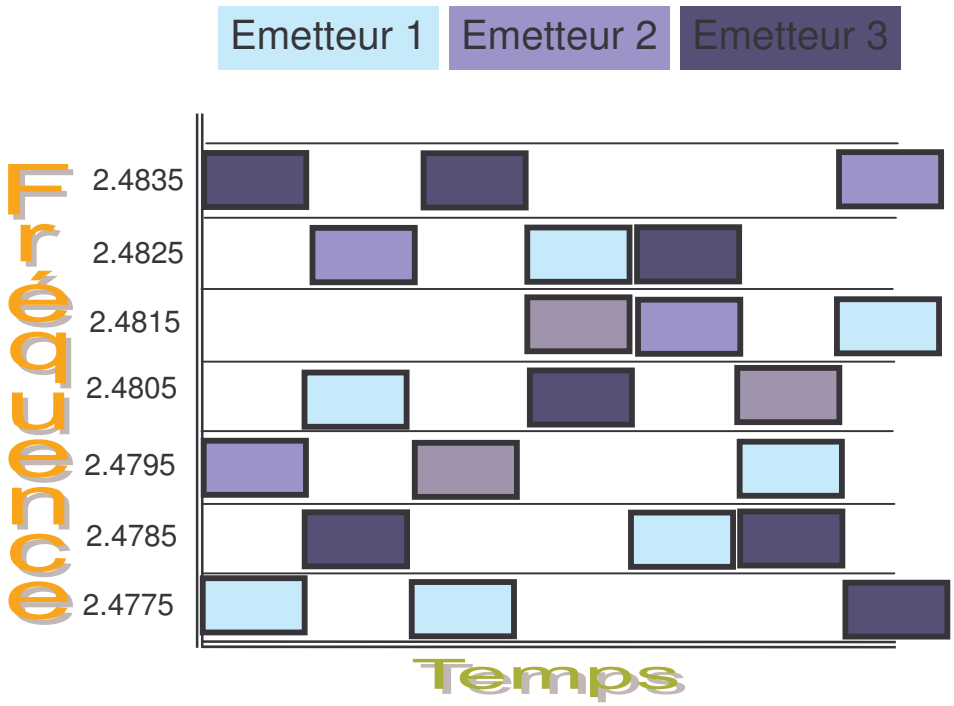
synchronisation. Ainsi, les équipements radio participant à une transmission utilisant FHSS doivent utiliser la même séquence de saut de fréquence pour pouvoir communiquer.

Historiquement, le FHSS a été inventé dans un but militaire pour empêcher l'écoute des transmissions radio. Le FHSS était un élément de sécurité car un équipement ne connaissant pas la combinaison de saut de fréquence ne pouvait ni écouter la communication, ni la localiser la fréquence ou le signal est émis car le temps d'émission était trop court.

L'utilisation de FHSS dans Bluetooth permet :

- Une synchronisation parfaite entre l'émetteur et le récepteur car ils sont obligés d'utiliser la même séquence de sauts pour communiquer (nous verrons comment se déroule cette synchronisation ultérieurement)
- D'émettre à plusieurs simultanément car en utilisant des combinaisons de saut de fréquences différentes, les fréquences sont ainsi partageables.
- De limiter les interférences (collisions) car les fréquences ne sont plus polluées.

Schématisons une transmission pour se visualiser le FHSS :



Comme on le remarque sur ce diagramme, 3 équipements Bluetooth peuvent émettre simultanément en utilisant les différents canaux de la bande de fréquence 2.400-2.483 Ghz.

IV.2. La couche bande de base (baseband)

La bande de base (ou baseband en anglais) est également gérée au niveau matériel.

C'est au niveau de la bande de base que sont définies les adresses matérielles des périphériques Bluetooth (équivalent à l'adresse MAC d'une carte réseau). Cette adresse est nommée BD_ADDR (Bluetooth Device Address) et est codée sur 48 bits. Ces adresses sont gérées par la IEEE Registration Authority.

C'est également la bande de base qui gère les différents types de communication entre les appareils. Les connexions établies entre deux appareils Bluetooth peuvent être synchrones ou asynchrones. La bande de base peut donc gérer deux liens de connexions :

- Les liaisons SCO (Synchronous Connection-Oriented) ;
- Les liaisons ACL (Asynchronous Conection-Less) ;
- Les liaisons de base.

IV.2.1. Les connexions SCO (Synchronous Connection-Oriented link)

Ce type de connexion permet une transmission bidirectionnelle. Une connexion SCO fonctionne en mode "Temps réel", c'est-à-dire qu'il n'y a pas de retransmission possible. C'est ce type de connexion qui est utilisé pour la transmission de voix.

Bluetooth utilise dans ce cas des créneaux réservés afin de réduire au maximum le délai. Il est alors possible d'atteindre un débit de 64Kb/s, sachant qu'un maître peut gérer jusqu'à 3 liens de ce type.

IV.2.2. Les connexions ACL (Asynchronous Conection-Less)

C'est ce type de connexion qui est utilisé pour échanger des données. Avec les connexion ACL, il est possible d'effectuer un broadcast et d'obtenir des débits de 723.2 Kbps en sortie et un débit de 57.6 Kb/s en entrée.

IV.2.3. Les liaisons de base

Ce type de liaison est utilisé pour la gestion des connexions au sein du réseau bluetooth (piconet expliqué dans la partie suivante).

Les paquets ont alors la forme suivante :

Code d'accès 72 bits	Entête 54 bits	Données 0 à 2745 bits

L'utilisation exacte de ces champs est présentée dans la dernière partie de ce document destiné à l'accès au support. Brièvement :

- Le code d'accès permet la synchronisation des composants Bluetooth.
- L'entête stocke le numéro de paquet, l'adresse source et destination, le type de paquet, le CRC…

IV.2.4. Les différentes topologies d'un réseau Bluetooth

Les liaisons de base gérée par la couche bande de base permettent de gérer les connexions au sein d'un réseau Bluetooth. Dans cette partie, nous allons vous présenter les différentes les piconet et les scatternet

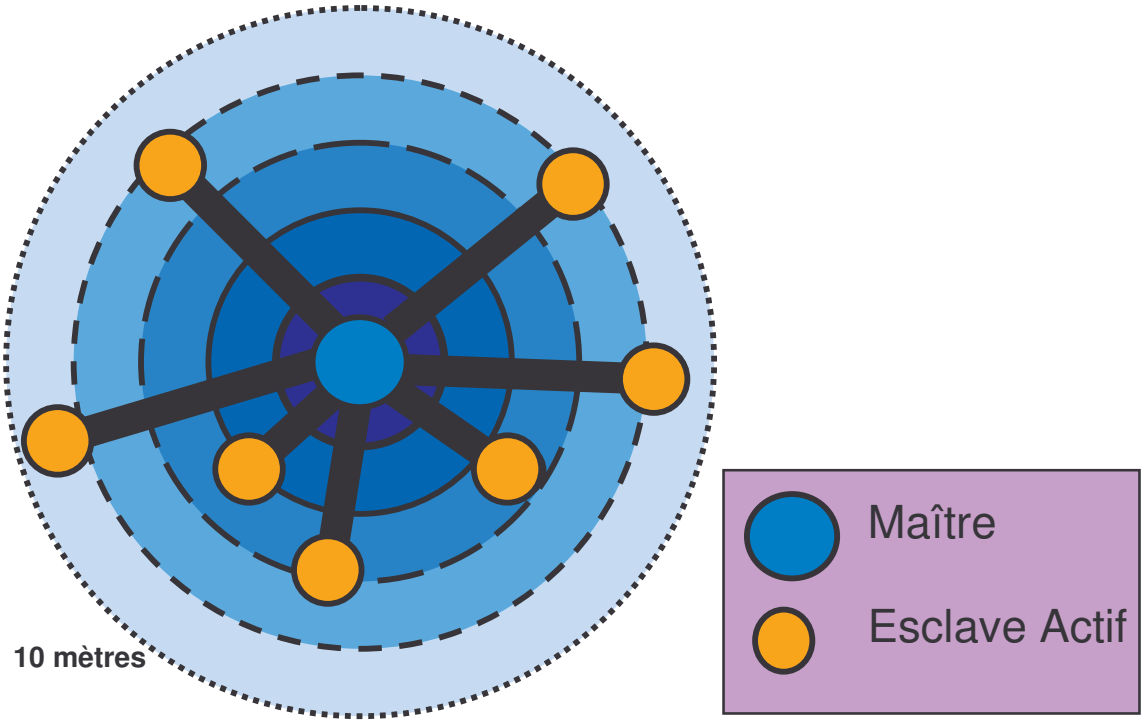
IV.2.4.1. Les Piconets

Un piconet est un réseau qui se crée de manière instantanée et automatique quand plusieurs périphériques Bluetooth sont dans un même rayon (10 m).

Ce réseau suit une topologie en étoile : 1 maître / plusieurs esclaves. Un périphérique maître peut administrer jusqu'à 7 esclaves actifs ou 255 esclaves en mode parked (=inactif).

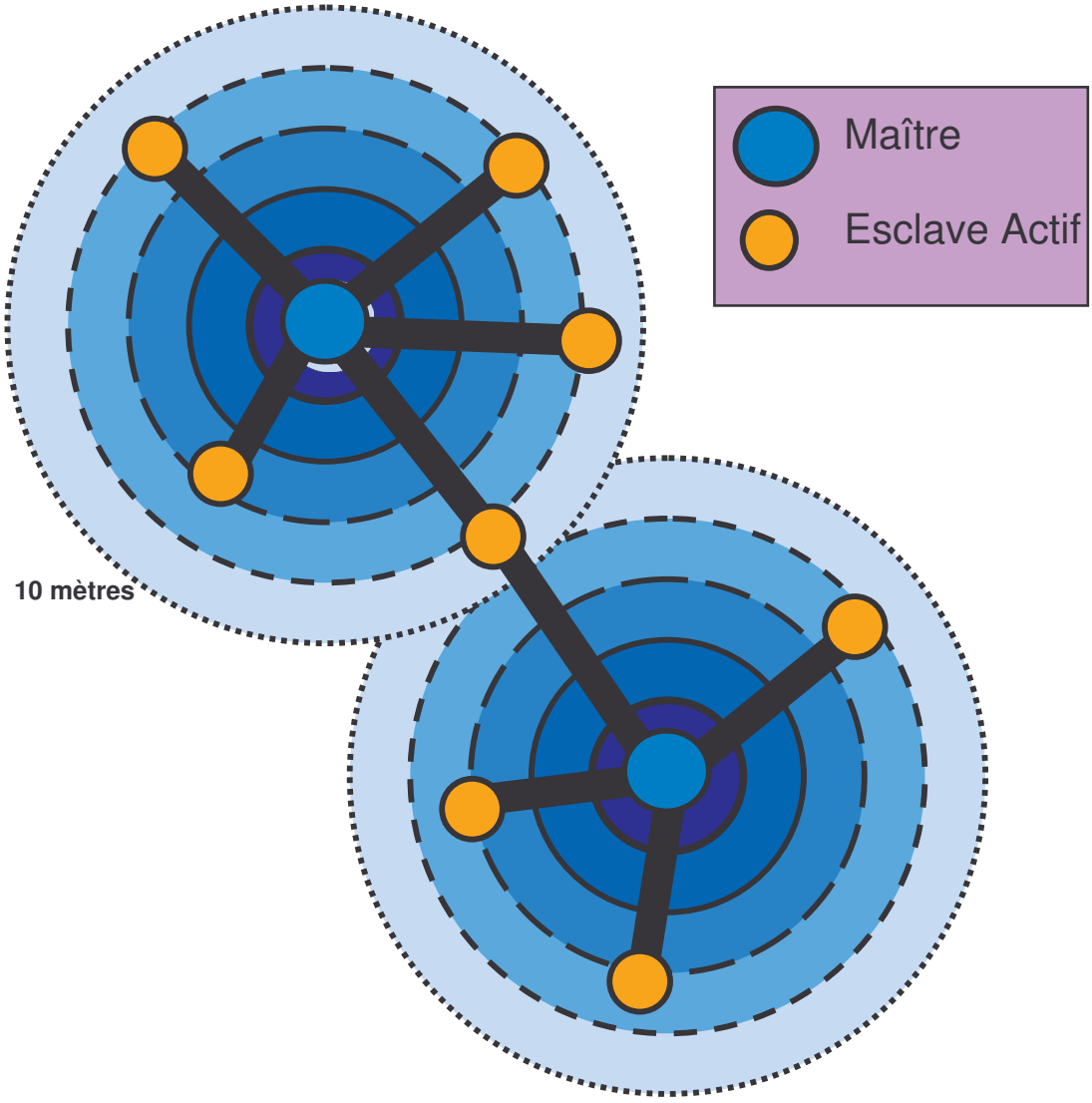
La communication est directe entre le maître et un esclave. Les esclaves ne peuvent pas communiquer entre eux.

Tous les esclaves du piconet sont synchronisés sur l'horloge du maître. C'est le maître qui détermine la fréquence de saut de fréquence pour tout le piconet.



IV.2.4.2. Les Scatternets

Les Scatternets sont en fait des interconnexion de Piconets (Scatter = dispersion). Ces interconnexions sont possibles car les périphériques esclaves peuvent avoir plusieurs maîtres, les différents piconets peuvent donc être reliés entre eux.



IV.3. La couche Link Manager

Cette couche est gère la supervision des différentes connexions, de l'authentification des appareils, et du chiffrement. Il gère également les mises en veille des différents appareils. Ce gestionnaire de liaisons qui implémente les mécanismes de sécurité comme :

- l'authentification,
- le pairage,
- la création et la modification des clés,
- le cryptage.

IV.4. La couche L2CAP (Link Layer Control & Adaptation)

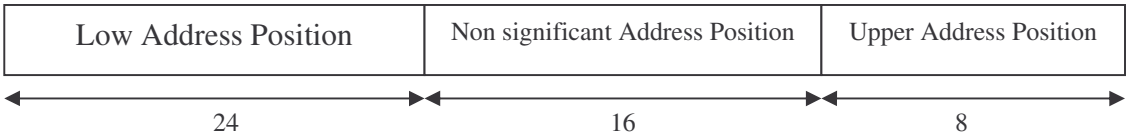
Cette couche permet l'adaptation des protocoles supérieurs au réseau Bluetooth. Elle comporte un mécanisme permettant d'identifier le protocole de chaque paquet envoyé pour permettre à l'appareil distant de passer le paquet au bon protocole, une fois celui-ci récupéré.

Cette couche supporte la segmentation et le réassemblage, et le multiplexage de protocole

V. ACCÈS AU MEDIUM

V.1. Les différents types d'adresse des dispositifs

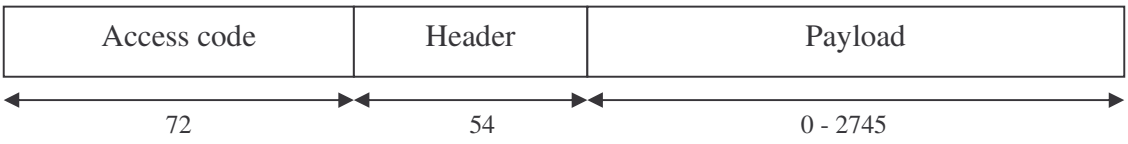
- **DBA (ou DB_ADDR)** : « Bluetooth Address Device » est une adresse unique pour chaque équipement Bluetooth. C'est l'équivalent de l'adresse MAC d'une carte réseaux. Elle est codée sur 48 bits comme montre le schéma ci-dessous :



- **AMA (ou AM_ADDR)** : « Active Member Address » est l'adresse d'un esclave dans le piconet. Elle est codée sur 3 bits dont l'adresse « 000 » est réservée pour le broadcast. Il peut donc avoir 7 esclaves au maximum dans un piconet.
- **PMA (ou PM_ADDR)** : « Parked Member Address » est l'adresse d'un esclave lorsqu'il se trouve à l'état parqué. Elle est codée sur 8 bits, donc il ne peut y avoir que 255 esclaves parqués au maximum.
- **ARA (ou AR_ADDR)** : « Access Request Address » est l'adresse de demande d'accès utilisé par les esclaves parqués. Cette adresse n'est pas nécessairement unique.

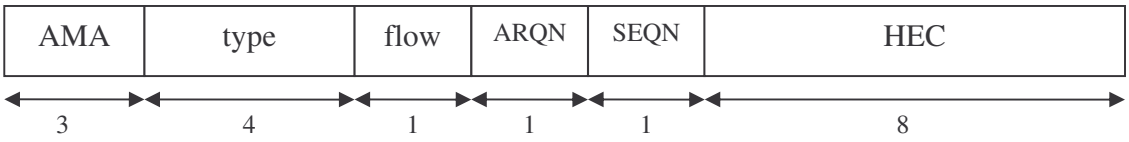
V.2. Format du paquet bluetooth :

V.2.1. Structure



Le paquet Bluetooth est composé de 3 différentes parties :

- **« Access code » (68/72 bits)** : Il identifie le piconet maître et il est utilisé pour la synchronisation, le « paging » et la recherche. Il y a trois types de code d'accès :
    - *CAC (Channel Access Code)* : Ce code de canal est identifie un unique piconet.
    - DAC *(Device Access Code)* : Le code de composant est utilisé durant pagination (« page » et « page scan ») et ses réponses. Ce code dérive de la BDA..
    - *IAC (Inquiry Access Code)* : Ce code est utilisé dans les procédures de recherche de composant. Il y a deux type de IAC :
        - GIAC (Generic IAC) : Utilisé par défaut pour rechercher tous les périphériques Bluetooth dans la zone. (0x9E8B33)
        - DIAC (Dedicated IAC) : Utilisé pour rechercher un type spécifique de composant.
- « **Header » (54 bits)** : L'entête est codé sur 54 bits. Il s'agit en fait d'une séquence de 18 bits répétés trois fois. Cette séquence est composée de six champs :



    - *AMA (3 bits)* : L'adresse actif de l'esclave. 0 pour le broadcast et 1 à 6 pour le périphérique.
    - *Type (4 bits)* : SCO, ACL, NULL, POLL / type de FEC / durée du payload
    - *Flow (1 bit)* : Contrôle de flow pour signaler que la mémoire tampon est pleine
    - *ARQN (1 bit)* : Indication de l'acquittement (ACK)
    - *SEQN (1 bit)* : Numéro de séquence
    - *HEC (8 bits)* : « Header Error Control »
- **« Payload » :** Données binaires utiles avec une zone de contrôle d'erreur de 16 bits.

V.2.2. Contrôle d'erreur

V.2.2.1. FEC

Les données peuvent être protéger par le code de correcteur d'erreur FEC (Forward Error Correction) :

- Le code FEC 2/3 nécessite 3 bits pour en protéger 2. (ex : 160b utiles pour 240b)
- Le code FEC 1/3 nécessite 3 bits pour en proteger 1. (ex. 80b utiles pour 240b)

Il s'agit d'un codage avec répétition de bits. Chaque en-tête d'un est toujours protéger par un FEC car elle est composé d'une séquence de 18 bits répétés 3 fois.

Cette protection réduit donc le débit utile mais, en contre partie, il permet la correction des paquets en erreur sur la liaison.

V.2.2.2. CRC

Pour les données, il s'ajoute au FEC une détection d'erreur CRC (Cyclic Redundancy Check) qui permet au récepteur de détecter les paquets en erreurs. Si le paquet est corrompu, le CRC peut demander un ARQ (Automatic Repeat Resquest) pour lui redemander ce paquet.

V.2.2.3. Automatic Repeat Resquest

Dans le protocole ARQ, les paquets sont retransmis jusqu'à ce qu'un acquittement soit reçu (ou le dépassement du délai de temps). Bluetooth emploie une confirmation rapide et non numérotée dans laquelle il emploie des confirmations positives et négatives en plaçant les valeurs appropriées dans l'en-tête. Si le délai de temps est dépassé, le paquet est perdu et la transmission est poursuivie avec la suite des paquets.

Dans la pratique,

- ARQN =1 si paquet précédent bien reçu et SEQN=SEQN+1 (SEQN = NOT SEQN)
- ARQN=0 si paquet précédent pas bien reçu. Cela arrive dans les cas suivants :
    - Le maître ne détecte pas le code d'accès. (paquet perdu)
    - HEC échoue
    - CRC échoue
    - SEQN = SEQN donc retransmission et ignore les paquets SEQN+1

Le broadcast n'a pas de ARQ.

V.2.2.4. Contrôle de Flux

Le protocole de bande de base recommande l'emploi de files d'attente de type FIFO dans les liens ACL et SCO pour la transmission et la réception. Le gestionnaire de lien remplit ces files d'attente et le contrôleur de lien les vident automatiquement.

Pour éviter que la file d'attente de réception soit pleine, ce qui provoquerait des pertes de paquet et de la congestion, on utilise un contrôle de flux. Une indication d'arrêt est transmise lorsque la queue est pleine. Elle est insérée par le contrôleur de lien du récepteur dans l'en-tête du paquet de retour. Lorsque l'émetteur reçoit l'indication d'arrêt, il bloque ses files d'attentes. Lorsque le récepteur est à nouveau prêt il envoie un paquet pour continuer la transmission.

V.3. Les différents types de paquet

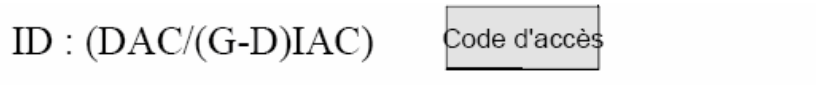
On peut distinguer 15 types de paquets différents que l'on peut séparer en trois catégories :

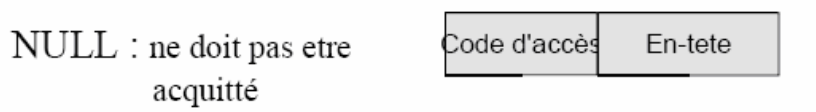
V.3.1. Paquet de contrôle

Ces paquets sont utilisés pour la gestion des connexions Bluetooth :

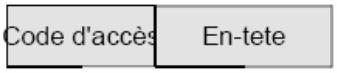
- ID (codé 0000) est utilisé pour le paging, la recherche, les réponses. Elle composé principalement composé d'un DAC ou IAC (GIAC ou DIAC)
- NULL (codé 0001) est utilisé pour faire une réponse de retour notamment lors d'une réponse à une requête POLL. Elle informe le maître sur le succès de la transmission, état de buffer, ect… Elle ne doit pas être acquittée.
- POLL (codé 0010) est idem que NULL mais il doit être acquitté. Il s'agit d'un paquet envoyé par le maître à l'esclave pour l'obliger a répondre même s'i n'a rien à dire.
- FHS (« Frequency Hopping Synchronisation » codé 0011) est utilisé pour synchroniser les esclaves lors de la mise en place du piconet. Ce paquet contient :
    - 240 bits de données utiles
    - L'adresse BDA de la source
    - L'horloge de la source
    - AMA donné par le maître a l'esclave
    - Des infos sur la période de paginations

Ces paquets et le paquet de type DM1 peuvent être utilisé sur les 2 types de liens SCO et ACL.





POLL : doit etre acquitté par NULL



V.3.2. Paquet SCO

Pour rappel, les liaisons à connexion synchrone orientée (SCO) se caractérise par :

- Transmission voix temps réel.
- Créneau réservé pour réduire au maximum le délai.
- Transmission point à point. Bidirectionnel.
- Il n'y a pas de retransmission du fait des contraintes temps réelles.
- Débit à 64 Kbit/s, pour être compatible avec les normes d'encodage.

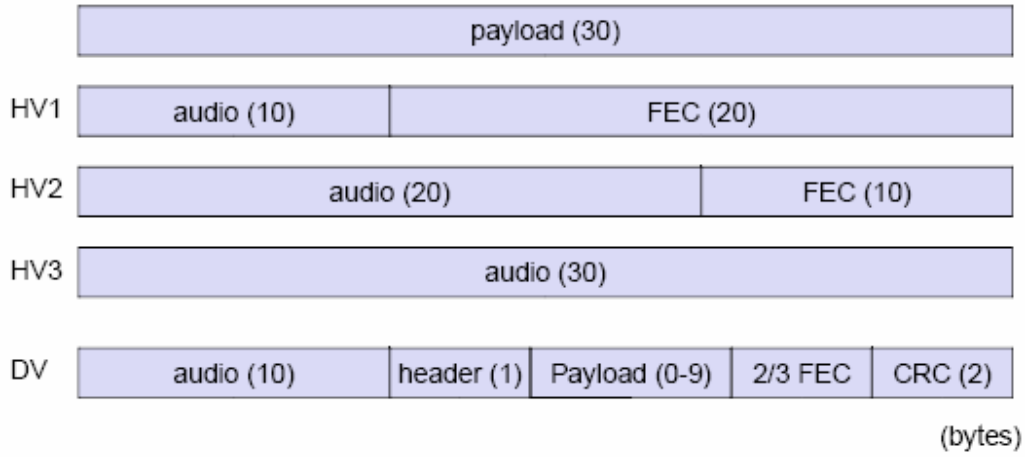
Il y a deux sous-catégories de paquet :

- Paquet HVx : « High quality Voice » sans correction d'erreur
- Paquet DV : « DataVoice» porte a la fois les données et la voix. Le ratio voix-donnée est d'environ 1/3 pour 2/3. La partie voix n'a pas de correction d'erreur et n'est jamais retransmise alors que la partie donnée est traitée séparément, on peut donc la retransmettre comme un paquet « standard ».

C'est pourquoi on peut remarquer que ces paquets n'on pas de CRC car elle ne nécessite pas de retransmission sauf pour le type DV.

Type	En-tête (octet)	Données Utiles (octet)	FEC	CRC	Débit max. symétrique (Kbit/s)
HV1	Aucun	10	1/3	Non	64
HV2	Aucun	20	2/3	Non	64
HV3	Aucun	30	Aucun	Non	64
DV	1 (pour les données)	10 + 0-9 D	2/3 sur les données	Oui sur les données	64 + 57,6 D

Voici la représentation de la charge utile par type de paquet SCO :



V.3.3. Paquets ACL

Pour rappel, les liaisons à connexion asynchrone (ACL) se caractérise par :

- Conçu pour l'échange de données.
- Le broadcast est possible.
- Schéma optionnel de retransmission en cas de paquets en erreurs (ARQ).
- Classification des paquets suivant la liaison.

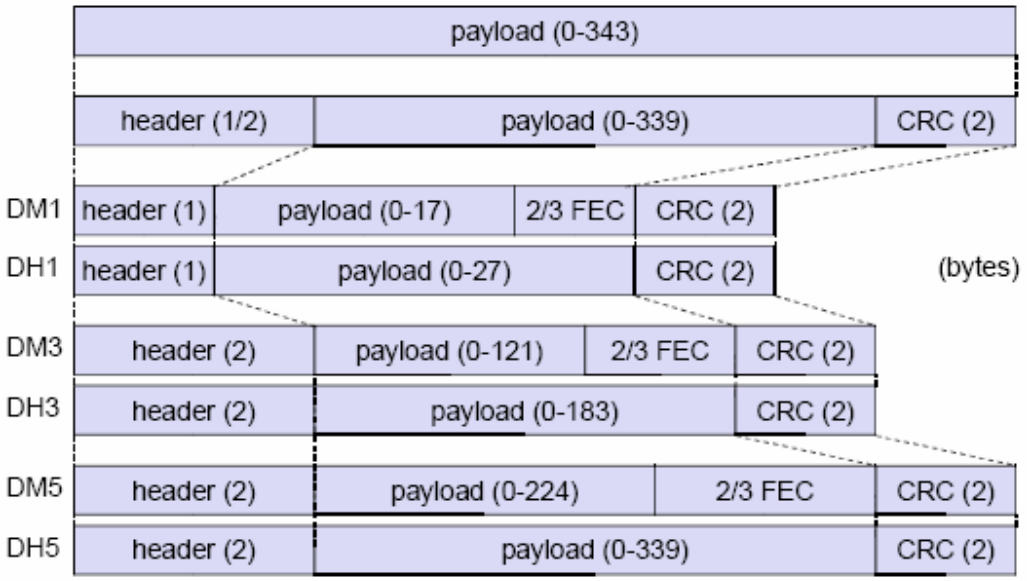
Il y a deux sous-catégories de paquet :

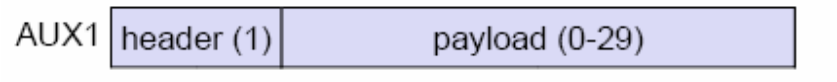
- Paquet DMx : « Data Medium » avec un encodage permettant la correction des erreurs en ligne
- Paquet DHx : « Data High » sans correction d'erreur, ce qui donne un débit effectif plus élevé.

Nb Slot	Type	En-tête (octet)	Données Utiles (octet)	FEC	CRC	Débit max. symétrique (Kb/s)	Débit max asymétrique (Kb/s)	
							descendant	montant
1	DM1	1	0-17	2/3	Oui	108.8	108.8	108.8
1	DH1	1	0-27	Non	Oui	172.8	172.8	172.8
3	DM3	2	0-121	2/3	Oui	258.1	387.2	54.4

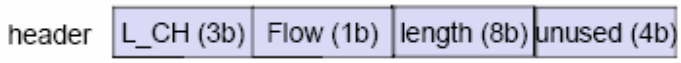
3	DH3	2	0-183	Non	Oui	390.4	585.6	86.4
5	DM5	2	0-224	2/3	Oui	286.7	477.8	36.3
5	DH5	2	0-339	Non	Oui	433.9	**723.2**	**57.6**
1	AUX1	1	0-29	Non	Non	185.6	185.6	185.6

Voici la représentation de la charge utile par type de paquet ACL :





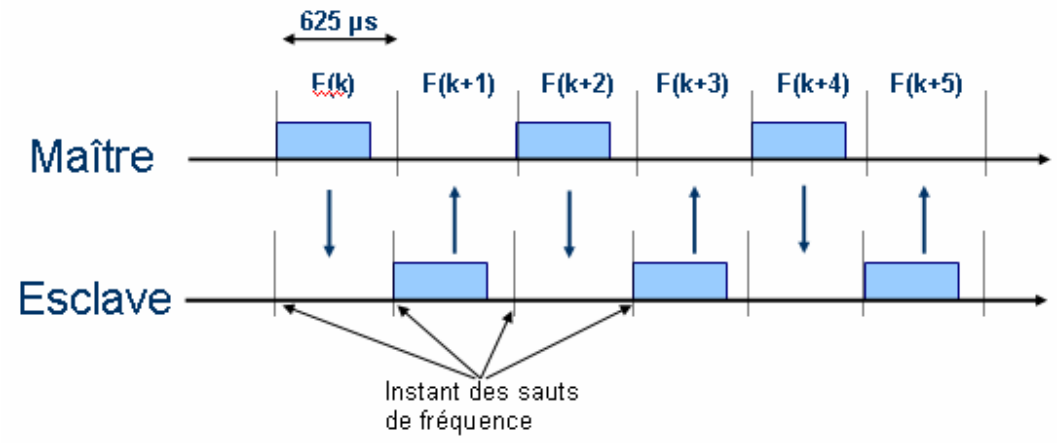
Et voici la structure de l'entête utilisé dans le payload :



V.4. Illustration du canal de communication

V.4.1. Communication single-slot

Le canal de communication est divisé en time slots numérotés d'une durée de 625µs chacun. Le TDD (Time Division Duplex) est utilisé, c'est à dire que les unités Maître et Esclaves transmettent alternativement (une unité Maître transmet dans les slots pairs et les unités Esclaves dans les slots impairs).



Les transmissions effectuées par les unités Bluetooth sont effectuées par paquets. Un paquet correspondant aux données transmises et reçues par les différentes entités Bluetooth.

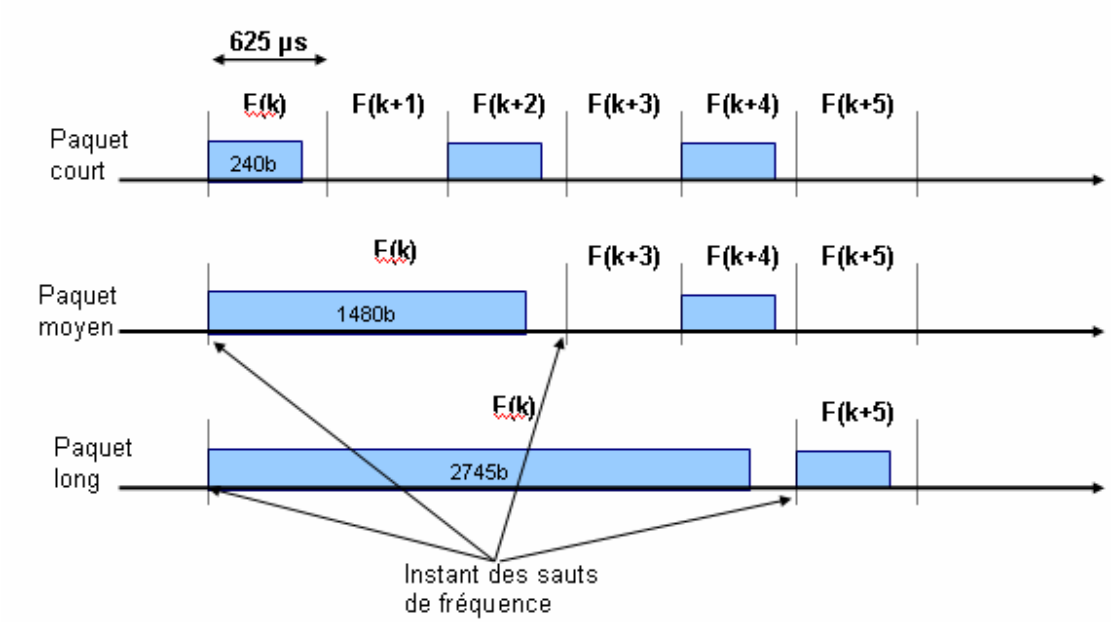
Chaque slot pour une communication maître esclave est systématiquement suivie par un slot esclave maître. Un esclave est uniquement autorisé à émettre dans un slot donné si le maître l'a adressé dans le slot précédent.

Toute communication directe entre esclaves est impossible, ils doivent passer par le maître du piconet. Le maître gère l'ordonnancement des esclaves selon l'algorithme Round Robin.

V.4.2. Communication multi-slot

Lorsqu'un paquet a une taille de 1 slot on parle de transmission/réception single slot, et lorsqu'un paquet possède une taille supérieure à 1 time slot (3 ou 5 slot), on parle de Multi-slot.

Le saut de fréquence appliqué au paquet est celui du premier slot de ce paquet.



Bluetooth peut donc utiliser 3 types de paquets :

- des paquets de données courts : 1 time-slot, 240 bits au maximum
- des paquets de données moyens : 3 time-slot, 1480 bits au maximum
- des paquets de données long : 5 time-slot, 2745 bits au maximum

Le débit peut donc varier selon les types paquets utilisés dans un sens puis dans l'autre.

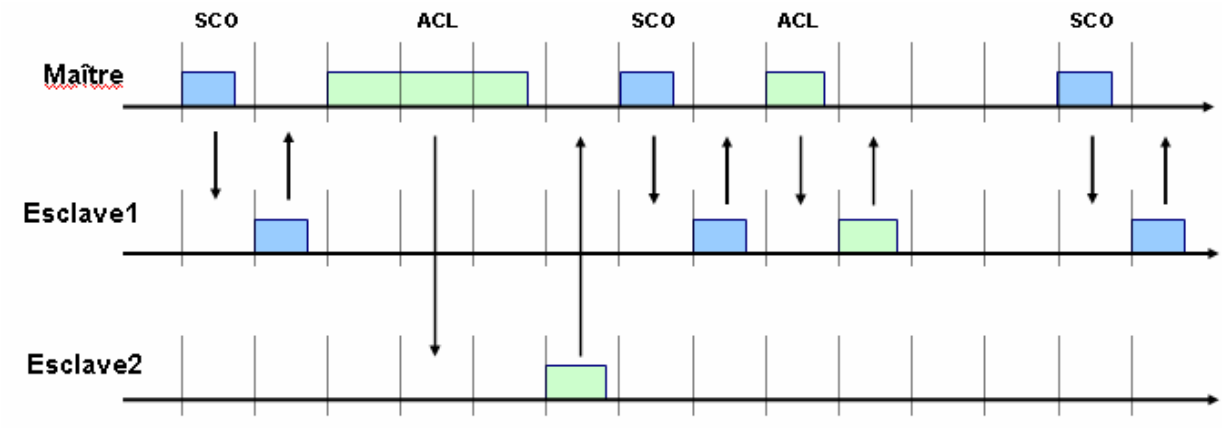
Par exemple :

- Paquet long, court dans l'autre sens : D1 = 2745 bits / 6 * 625 us = 732 kbits/s D2 = 240 bits / 6 * 625 us = 64 kbits/s
- Paquet long dans les deux sens D1 = 2745 bits / 10 * 625 us = 439 kbits/s = D2

Le débit maximal de 732Kbits/s n'est obtenu que s'il s'agit de la connexion asymétrique avec un paquet long d'un sens et un paquet court dans l'autre sens.

Voici une illustration d'une communication le maître échange des données avec 2 esclaves du piconnet :

- Esclave 1 un lien SCO et un lien ACL
- Esclave 2 un lien ACL



Les liaisons asynchrones

Le mode asynchrone privilégie un débit élevé dans une direction : 721 kb/s dans un sens, contre seulement 57,6 kb/s dans l´autre. La direction peut être fixée temporairement par l´utilisateur ou par l´application et implique la définition d´un maître et d´un esclave entre les périphériques communicants. C´est la solution qui sera retenue en principe dans le cadre d´un accès à Internet via Bluetooth si les téléchargements sont plus fréquents que les uploads.

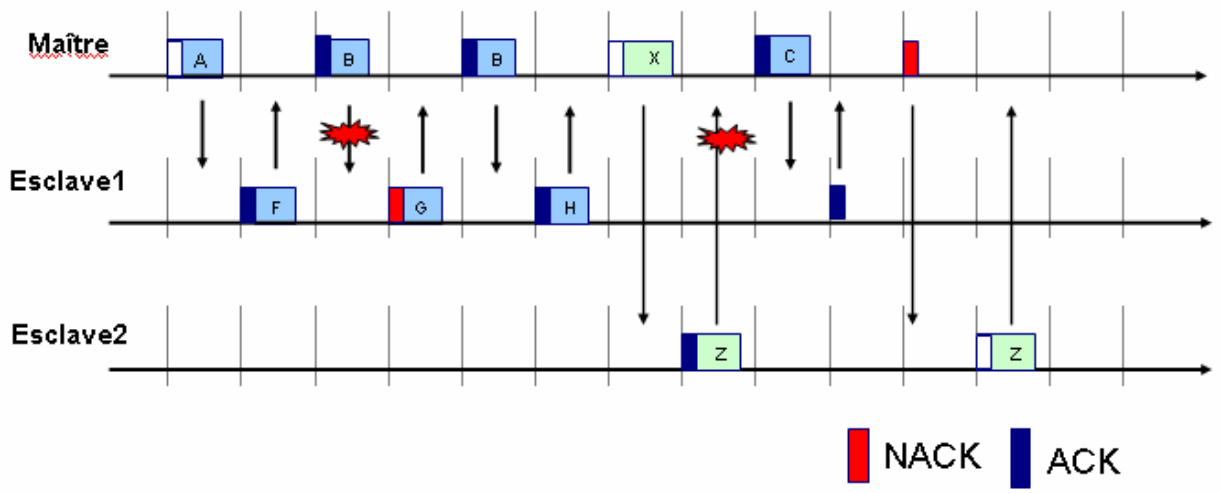
De même, cette solution sera également celle préférée pour les imprimantes. Il est en effet inutile de réserver une large bande aux communications de l´imprimante vers le poste de travail. En revanche, les communications asynchrones peuvent présenter des discontinuités. Elles ne sont donc pas adaptées à la transmission de parole, de vidéo (même si le débit actuel, limité à 1 Mbits/s ne permet pas d´envisager de toute façon de solution plein écran), ou de musique.

Les canaux voix/données

Enfin, Bluetooth propose trois canaux dits "vocaux" synchrones. Bidirectionnels, ils possèdent un débit de 64 Kb/s.

Plus clairement, le débit de 64 Kb/s est assuré simultanément en envoi et en réception et s´avère donc particulièrement adapté à la transmission de la voix ou de tout fichier numérique devant être reconstitué en temps réel : communication téléphonique, MP3, etc.

V.4.3. Retransmission automatique



La transmission du paquet B vers l'esclave 1 a été perturbé, l'acquittement n'a pas eu lieu (NACK) et le paquet est renvoyé dans le slot suivant : c'est la retransmission automatique.

La transmission du paquet Z envoyé par l'esclave 2 vers le maître a été perturbée. Dans ce cas, le maître averti l'esclave dès qu'il est disponible pour recevoir une nouvelle retransmission du paquet.

En résumé, la technique temporelle synchronisée consiste :

- Temps divisé en tranches de longueur égale = slots
- 1 Slot = temps de transmission

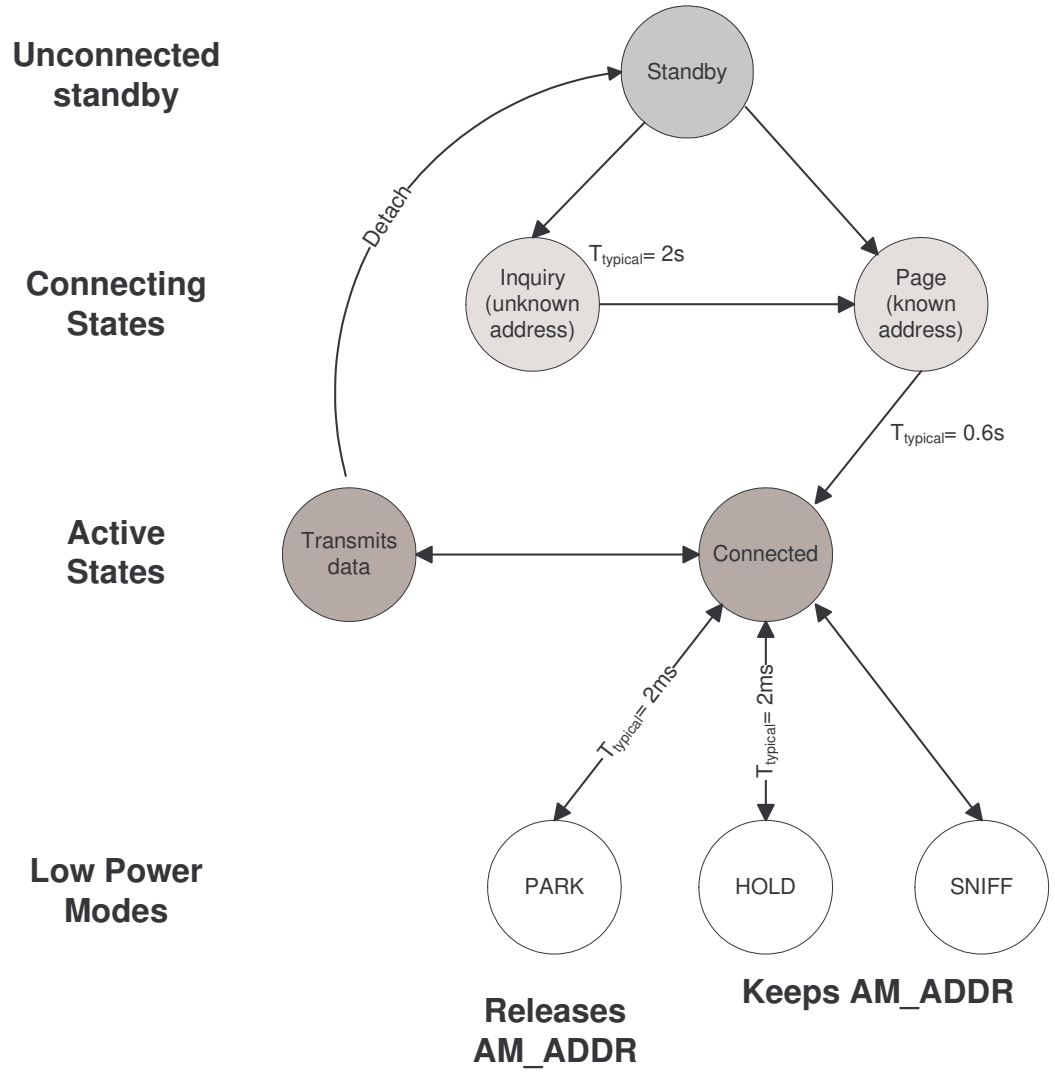
Le temps est découpé en slots :

- 1600 slots/s
- 1 slot : 625 microsecondes de long
- Un terminal utilise 1 fréquence sur 1 slot, puis, par un saut de fréquence (Frequency Hop), il change de fréquence sur la tranche de temps suivante, etc.
- Un client Bluetooth utilise de façon cyclique toutes les bandes de fréquence.
    - Les clients d'un même piconet possèdent la même suite de sauts de fréquences
    - Lorsqu'un nouveau terminal veut se connecter, il doit commencer par reconnaître l'ensemble des sauts de fréquences pour pouvoir les respecter
- Une communication s'exerce par paquet (1, 3 ou 5 slots) : le saut de fréquences a lieu à la fin de la communication d'un paquet

V.5. Etats des terminaux Bluetooth

Le contrôleur Bluetooth fonctionne dans 2 états principaux : « stand-by » et « connexion ». Il y a 7 états subsidiaires qui sont utilisés pour ajouter des esclaves ou créer des connexions dans le Piconet. Ces sous états sont : page, page scan, inquiry, inquiry scan, master response, slave response et inquiry response. Un périphérique bluetooth peut prendre les quatre états suivants : **Active, Hold, Sniff** and **Park**.

Voici le schéma des différents états que peut prendre un périphérique Bluetooth :



V.5.1. Etats non-connectés

V.5.1.1. Standby, radio éteinte et mode low-power

L'état **Standby** est l'état par défaut de basse consommation pour un périphérique Bluetooth. Seule l'horloge native est actif et il n'y a aucune interaction avec les autres dispositifs. En mode connecté, le maître et l'esclave peuvent s'échanger des paquets en utilisant le code d'accès du canal du maître ainsi que son horloge.

Dans ce mode, une unité non connectée « écoute » les messages périodiquement toutes les 1.28 secondes. A chaque fois qu'une unité rentre en mode actif, celle-ci écoute un ensemble de 32 sauts de fréquences qui lui est propre.

V.5.1.2. Inquiry, identification et écoute

- **Inquiry**

Lorsqu'un périphérique désire découvrir les nouveaux dispositifs, il passe en état « inquiry » , où il envoie des paquets de broadcast « inquiry », contenant un IAC, à tous les périphériques dans sa zone. Il l'envoie en utilisant le « inquiry hopping sequence » c'est-à-dire qu'il l'envoie au 32 fréquence d'éveil. Il obtient sa synchronisation et le code d'accès du canal par FHS.

Le dispositif peut alors recevoir des réponses a ces « inquiries », mais il ne devra pas acquitter ces paquets de réponse.

Le message INQUIRY est utilisé afin de communiquer avec des équipements dont on ne connaît pas l'adresse (par exemple des imprimantes ou des fax publics).

- **Inquiry Scan**

Lorsqu'un périphérique désire recevoir des paquets « inquiry », il entre en mode « inquiry scan ». Il utilise le « inquiry hopping sequence » c'est-à-dire qu'il écoute successivement sur les 32 fréquences d'éveil.

Il s'agit d'une écoute active de DIAC et GIAC. S'il reçoit un « inquiry », il renverra un FHS.

- **Page**

Une connexion est établie par un message de type PAGE si l'adresse de l'unité à connecter (unité Esclave) est connue.

Dans l'état initial PAGE, l'unité Maître envoie un train de 16 messages identiques de paging sur 16 différents sauts de fréquences spécifiques à l'unité pagée (Esclave). Ce train de message couvre la moitié de la séquence de sauts de fréquences que l'unité Esclave écoute en mode

STANDBY et il est répété 128 fois, ce qui correspond à 1.28 s. Si aucune réponse n'est reçue après ce délais, le Maître retransmet le même train de message de paging dans les 16 sauts de fréquences restant de la période d'écoute de l'unité Esclave.

Le délai maximum pour que l'unité Maître atteigne une unité Esclave est donc de deux fois 1.28 s c'est à dire 2.56 s.

- **Page scan**

Lorsqu'un périphérique désire recevoir des paquets « page », il entre en mode « page scan ».

V.5.2. Etats connectés

Dans le mode actif, l'unité Bluetooth participe activement sur le canal. Le maître organise les transmissions de base sur la demande de trafic ainsi qu'en fonction des esclaves. En plus, il supporte des transmissions régulières pour que les esclaves restent synchronisés sur le canal. Les esclaves actifs écoutent durant les « slots » maître à esclaves pour les paquets. Si un esclave actif n'est pas adressé, il peut dormir jusqu'à la prochaine transmission du maître.

Représentation du mode actif :

Pour les unités connectées, trois modes d'économie d'énergie peuvent être utilisés si aucune donnée ne doit être transmise :

- **Mode HOLD**

L'unité Maître peut forcer les unités Esclaves à passer en mode HOLD. Dans ce mode, il n'y a plus que l'horloge interne qui fonctionne. Les unités Esclaves peuvent aussi demander à passer en mode HOLD. Le transfert de données ne reprendra que lorsque l'unité aura quittée le mode HOLD. Le mode HOLD est typiquement utilisé dans le cas de connexions avec plusieurs piconets, ou encore lorsque les données ne sont pas envoyées très fréquemment. C'est idéal pour les événements non périodiques et plutôt utiles pour le maître qui aurait besoin d'établir des liens supplémentaires ; Le mode démarre sur une valeur d'horloge prédéfinie. Représentation mode HOLD :

- **Mode SNIFF**

Dans ce mode, un dispositif esclave écoute le Piconet à un taux réduit, de ce fait réduisant son coefficient d'utilisation. La périodicité est programmable et dépend de l'application ; Il s'agit donc d'une opération répétitive qui est plutôt adapté pour les événement périodique, pour économiser l'énergie des liens ACL (si une certaine latence est toléré), et procure une certaine flexibilité dans les trafics encombré. Représentation mode SNIFF :

- **Mode PARK**

Dans ce mode une unité est toujours synchronisée au piconet mais ne participe pas au trafic. Ces dispositifs ont rendu leur adresse (AM_ADDR) et écoutent occasionnellement le trafic du maître pour se re-synchroniser et vérifier les messages de diffusion généraux (Broadcast). Une adresse membre de parking lui est assigné (PM_ADDR). Une adresse de requête lui est aussi assignée (AR_ADDR) que le maître utilisera pour le faire sortir du mode « park ».

L'usage premier du mode park est lorsque plus de 7 esclaves sont attchés au piconet. Il est périodique comme le mode SNiFF mais d'une strucure plus rigide car il augmente sa charge, mais permet une meilleur économie d'energie. Il a comme travail supplémentaire de vérifier régulièrement si un esclave est toujours présent. L'esclave doit quitter le mode park pour pouvoir recevoir ou transmettre des données. Ce mode ne peut pas être utilisé pour les esclaves en lien SCO actif.

V.6. Comment fonctionne Bluetooth

V.6.1. Au départ

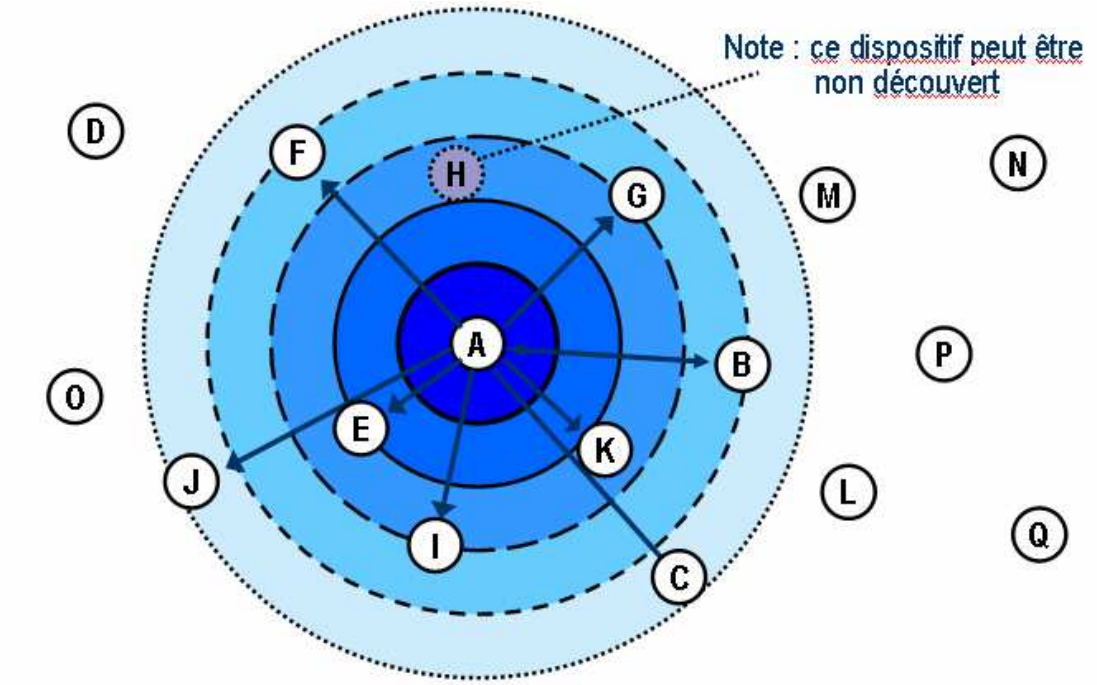
Initialement, les périphériques Bluetooth ne se connaissent qu'eux même et ils sont en mode StandBy. StandBy est un mode passif (non-connecté) où les périphériques cherchent la présence de transmission (« Inquiry » ou « Page Scans ») dans sa zone de couverture pendant 10ms toutes les 1.28 secondes pour voir si aucun des périphériques cherchent à communiquer. Ils écoutent successivement 32 « porteuses d'éveil » parmi les 79 fréquences.

Les états passifs sont occupants plus de la moitié de états Bluetooth et c'est le mécanisme clef pour réaliser un dispositif de très faible consommation. Parfois, un périphérique Bluetooth en mode Standby peut réduire sa consommation jusqu'à 98%.

Il est important de noter que les périphériques ne sont pas encore synchronisé. Tous les périphériques sont tous en train d'écouter à des temps différents et sur des fréquences différentes.

V.6.2. Découverte

Ce processus est différent du processus de pagination car elle possède moins de renseignements. Le master n'utilise pas l'ACCESS CODE, il transmet le GIAC (General Inquiry Access Code) ou le DIAC (Dedicate Inquiry Access code). Le GIAC permet d'extraire l'information sur les capacités des slaves, le DIAC lui informe sur des capacités plus spécifiques. Tous les éléments utilisent la même fréquence porteuse. Le slave répond à une requête avec ses paquets FHS après que le master lui aie attribuer un numéro propre de trame impaire. Les quantités de paquets FHS, reçues, permettent au master de se caler sur le slave et de commencer une communication, en effet les FHS permettent au master d'élaborer la BD_ADDR et donc de construire des paquets valides.



Le mode découverte (Inquiring) est un mode de fonctionnement qui permet de connaître les autres périphériques qui sont dans la zone de portée.

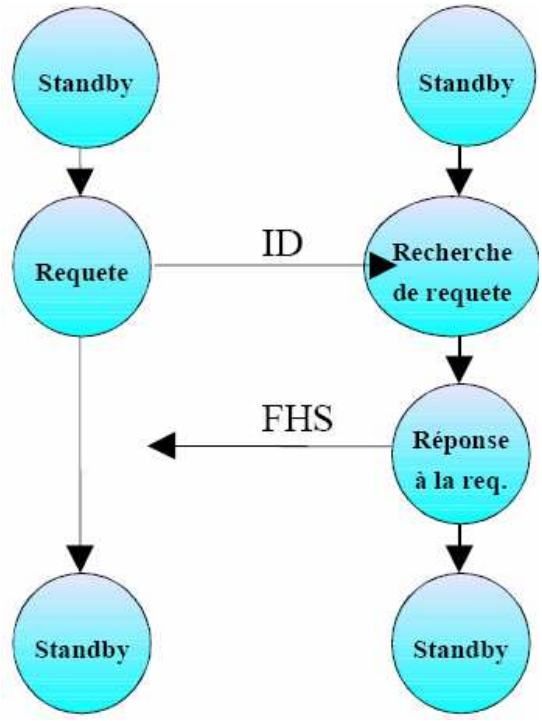
L' « inquiring » est la manière dont un dispositif de Bluetooth se renseigne sur d'autres dispositifs qui sont dans sa zone. Dans l'illustration au-dessus du noeud A exécute une procédure de pagination sur le « BT inquiry ID » (ID de Broadcast) et reçoit des réponses des dispositifs B, C, E, F, G, I, J, et K. Par ces réponses le dispositif A apprend l'identité de ces autres dispositifs.

Pendant la phase de découverte, le noeud A broadcast continuellement la commande de pagination en utilisant le « BT inquiry ID ». Ce qui l'identifie comme une requête « Inquiry ». Ces broadcasts sont diffusés au travers des 32 fréquences porteuses « Standby » où tous les dispositifs en mode Standby sont à l'écoute.

Après quelques secondes, chaque dispositif dans la zone aura reçu l' « inquiry » mais ils ne sont aucunement synchronisés. Par convention ces noeuds répondront avec un paquet standard FHS qui fournit leur identification de BT unique et leur horloge. Avec ces paramètres le noeud d'investigation peut effectuer des connections synchronisées de faible latence.

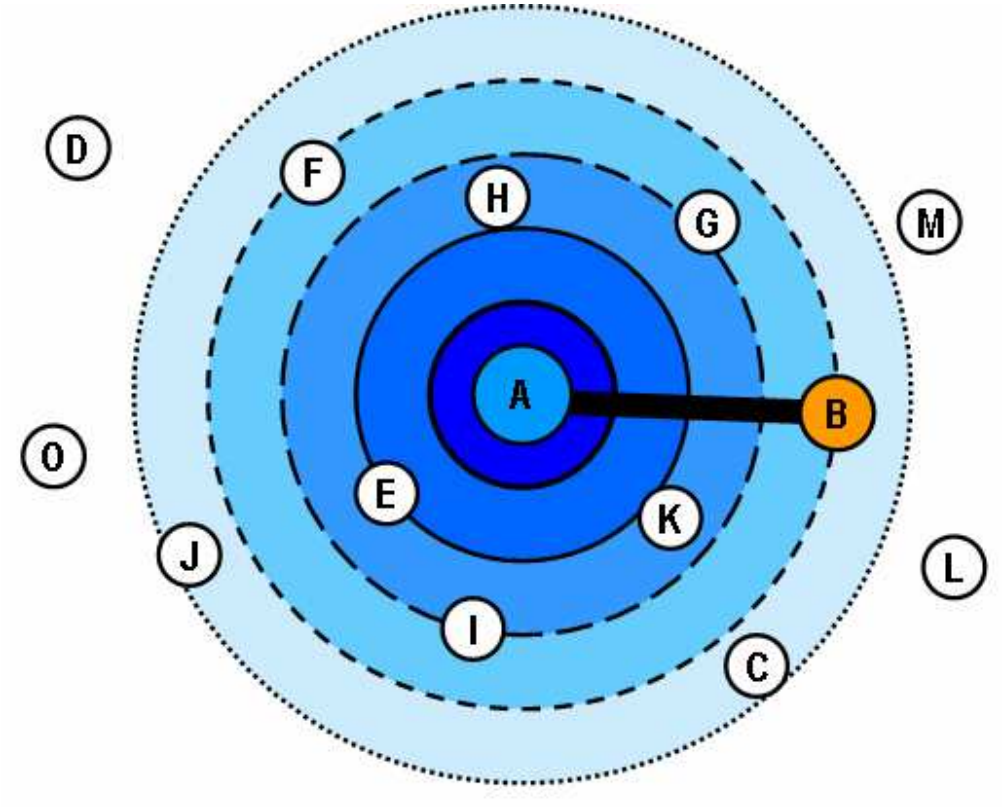
Ici le noeud H illustre un dispositif Bluetooth programmé pour rester anonyme (« Undiscoverable »). C'est une option qui suspend le « Inquiry scan » ainsi aucun dispositif ne peut le découvrir.

Cependant il faut noté que le nœud H continue à assurer la fonction « page scan ». On peut créer une connexion avec celui-ci en lui envoyant une requête de pagination directement à son Bluetooth Unique ID.



V.6.3. Créer un piconet

La création d'un piconet (Paging) permet de créer un lien Maître/Esclave.



Les connexions Maître/esclave en Bluetooth sont désignées sous le nom de Piconet.

Pour créer un piconet, le nœud A diffuse (ou broadcast) une commande de pagination avec un ID Bluetooth explicite (Ici nœud B). Cette ID utilisé a été récupéré plutôt dans la phase de recherche par la procédure « Inquiry » vu plus haut.

Tous les périphériques Bluetooth excepté le B ignorent cette commande de pagination simplement parce qu'elle ne leur est pas adressée. Lorque le nœud B répond, le nœud A enverra à son tour un paquet FHS et lui assignera un AM_ADDR dans le piconet.

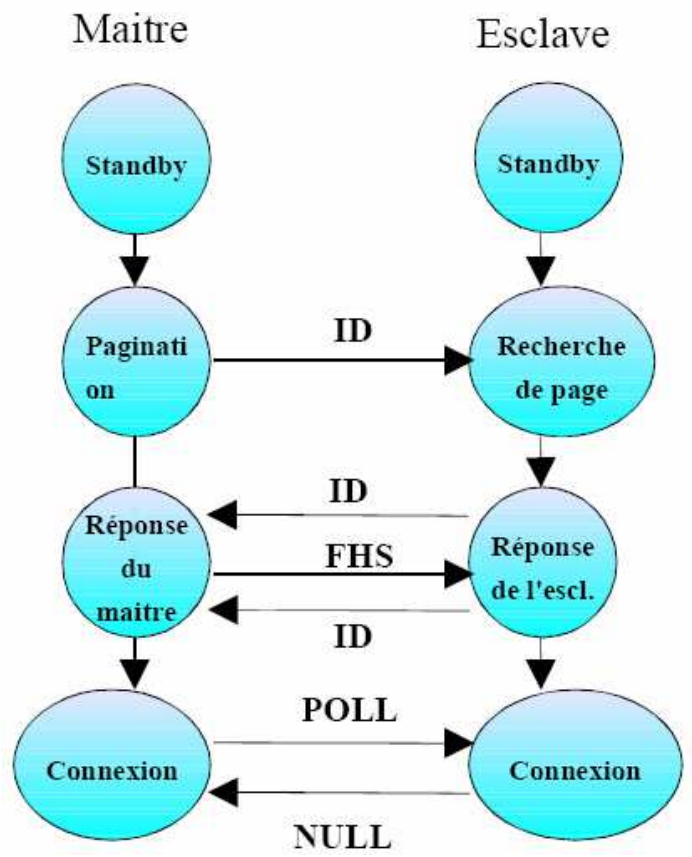
Le nœud A devient alors le maître du piconet car c'est lui qui a fait la requête de pagination, et son adresse définit la suite des sauts en fréquence que devront suivre les esclaves.

Le noeud B est maintenant en état actif et se mettra en écoute pour toutes autres commandes provenant du nœud A. Il doit se synchroniser sur les sauts de fréquence du nœud A et calé son horloge.

Le problème est que le maître ne sait pas sur laquelle des 79 fréquences porteuses se fixer. La solution est :

- Balayage des pages sur 32 fréquences porteuses.
- La BD_ADDR contient l'information sur la bonne fréquence à adopter, elle est connu par le master.
- Utilisation de l'horloge Bluetooth des éléments, on connaît ainsi le temps écoulé pendant la dernière connexion de deux unités. Cela a pour effet d'augmenter la vitesse d'élaboration de la page.

En effet le balayage de la page d'état est fait toutes les 2.56 s. Le slave consulte cette page toutes les 11.25 ms à fréquences porteuses variables. Pendant ce temps le master potentiel lance des paquets d'identification pour le slave potentiel sur deux fréquences porteuses différentes en utilisant les trames numérotées paires. Il écoute les réponses sur les trames impaires consécutives. La taille réduite de l'ACCESS CODE permet au dispositif électronique du master de switcher sur deux fréquences différentes en 625 micro s. Ainsi pendant 11,25 ms le master peut envoyer et recevoir les informations de 16 canaux différents. Ne sachant pas quand le slave se reconnecte à la page d'état, il réitère cette opération pendant 2.56s. Si le résultat n'est pas concluant, le master passe en revue 16 autres fréquences jusqu'à réponse ou timeout.



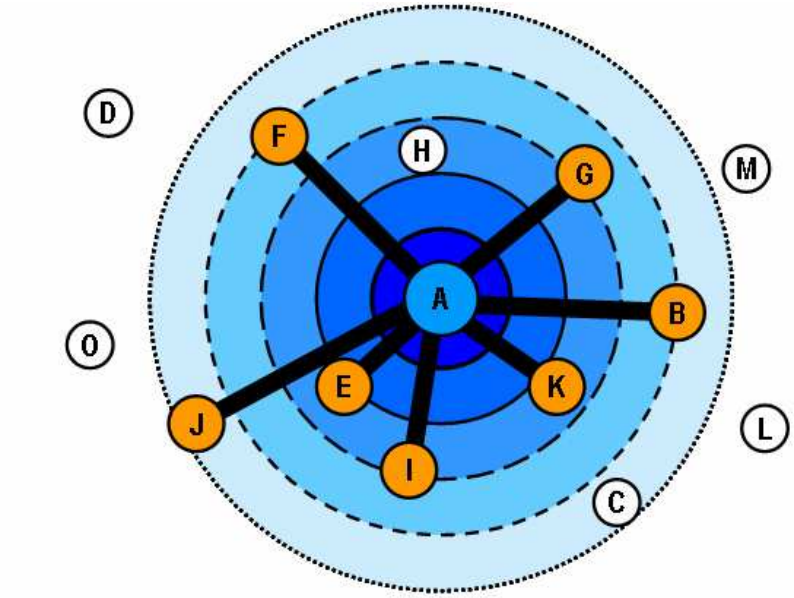
L'état de connexion débute avec un paquet POLL (scrutin) envoyé par le maître pour vérifier que l'esclave a commuté et s'est synchronisé sur le maître ainsi que sur la séquence du canal.

Il peut aussi arriver qu'un dispositif (esclave) désire se connecter à un Piconet sans y être invité par le maître. La procédure est presque la même que celle précédemment expliquée, mais voici dans les grandes lignes ce qu'il faut faire :

- Une adresse de connexion est réservée, le dispositif en tire le code d'accès
- Il écoute pendant un long moment la même fréquence
- Par les paquets de Broadcast envoyés par le maître il peut obtenir la séquence de saut du Piconet ainsi que les informations supplémentaires qu'il aurait besoin.

V.6.4. Etendre un piconet

Des Paginations successifs permettent d'attacher jusqu'à 7 esclaves actifs sur un même maître.



A travers des successions de commande de pagination, un maître peut s'attacher jusqu'à 7 esclaves actifs.

La limite est 7 esclaves car :

- AMA est codé sur 3 bits
- 000 est réservé au maître du piconet
- Le reste pour les esclaves.

CONCLUSION

Bluetooth est une technologie qui se caractérise par :

- un faible coût
- une faible consommation
- une taille réduite
- disponible dans le monde entier (bande ISM)
- ayant une bonne résistance aux interférences

Cependant Bluetooth est limité et les principaux reproches sont :

- Problèmes de compatibilité entre les puces provenant de divers industriels
- Débit faible
- Nombre de périphériques en réseau limité
- Technique de partage de l'interface radio peu apte à passer à des vitesses plus élevées (technique de polling interrogation/réponse)
- Concurrence de la norme IEEE 802.11

On compare souvent Bluetooth à des soi-disant concurrents, comme WiFi et d'autres du même type. Bluetooth convient parfaitement aux applications qu'on lui confie tout comme WiFi a les siennes. Les réseaux WiFi sont relativement performants au niveau du débit, par contre ils consomment plus d'énergie. Contrairement à Bluetooth qui a un débit relativement faible (1Mbits/s, WiFi ~11-54Mbits/s) mais par contre consomme peu d'énergie. Chaque technologie doit être utilisée dans le but premier pour laquelle elle a été conçue.

La spécification Bluetooth 1.2 a été publiée. Elle promet un débit 3 fois plus rapide pour une consommation 2 fois moindres. Ce qui sera toujours insuffisant pour faire du réseau (par ex.) mais sera encore plus performant pour les applications actuelles de faible consommation telles que les oreillettes pour téléphones portables et autres souris sans fil.