Chapitre 1

La technologie Bluetooth : paradigmes et ordonnancement

1.1 Introduction

Les réseaux Bluetooth ont retenu l'attention des chercheurs et des industriels durant ces dernières années parce qu'ils permettent des communications instantanées, à faible coût, entre des équipements hétérogènes. Cependant, l'évolution de ces réseaux pour intégrer les applications multimédia (voix sur IP, audio et vidéo à la demande, etc.) soulève de nombreux problèmes qui tournent autour de la dégradation des performances. L'un des problèmes à gérer, dans ce contexte, est le choix des techniques d'ordonnancement pouvant garantir la qualité de service.

Dans le reste de ce chapitre, nous décrivons, dans une première étape, les réseaux Bluetooth. Dans une seconde étape, nous présentons les solutions d'ordonnancement proposées dans la littérature pour ce type de réseaux. Pour ces solutions, nous nous concentrons sur le support simultané des contraintes degré d'importance et échéance de remise des messages ; contraintes dont nous nous intéressons pour répondre aux besoins de qualité de services des applications.

1.2 La technologie Bluetooth

Bluetooth est une technologie de réseau personnel sans fil WPAN (Wireless Personal Area Network) qui a été créée par Ericson en 1994. Ensuite, à partir de 1998, elle a été développée par l'organisation Bluetooth Special Interest Group (Bluetooth SIG). Cette organisation comporte des personnes de différentes sociétés dont font partie Ericsson, Intel, IBM, Nokia, Toshiba, 3Com, Lucent Technologies, Microsoft et Motorola.

Bluetooth s'appuie sur des spécifications ouvertes qui ont évolué au cours du temps. La plupart des produits Bluetooth sont compatibles avec les spécifications 1.0b et 1.1 [10]. Dans le but de supporter un champ plus vaste d'applications, une nouvelle version 1.2 des spécifications [11] a été approuvée en novembre 2003.

Cette nouvelle version est compatible avec les versions antérieures.

Bluetooth [10] correspond à une interface radio à portée faible (de l'ordre de quelques dizaines de mètres) permettant de remplacer le câblage reliant des équipements informatiques et/ou électroniques. Bluetooth vise une connexion sans fil avec une faible complexité, une faible consommation d'énergie et un faible coût. Cette technologie permet de transmettre des données et de la voix dans la bande de fréquence ISM (Industrial, Scientific and Medicine) entre 2,4 GHz et 2,48 GHz.

Bluetooth peut fournir une connexion point à point (uniquement deux unités Bluetooth) ou une connexion point à multipoint, comme le montre la figure 1.1. Pour une connexion point à multipoint, le canal est partagé entre plusieurs unités Bluetooth. Deux ou plusieurs unités, partageant le même canal, forment un piconet. Une seule unité d'un piconet se comporte comme maître et gère l'accès au canal alors que les autres unités sont des esclaves. Dans un seul piconet, nous pouvons avoir, au maximum, sept esclaves actifs.

Plusieurs piconets peuvent être interconnectés formant un scatternet (voir figure 1.1). Ces piconets ne sont pas obligatoirement à fréquences synchronisées puisque chacun a son propre canal. Cependant, les esclaves peuvent participer dans différents piconets. De même, le maître d'un piconet peut être un esclave dans un autre piconet.

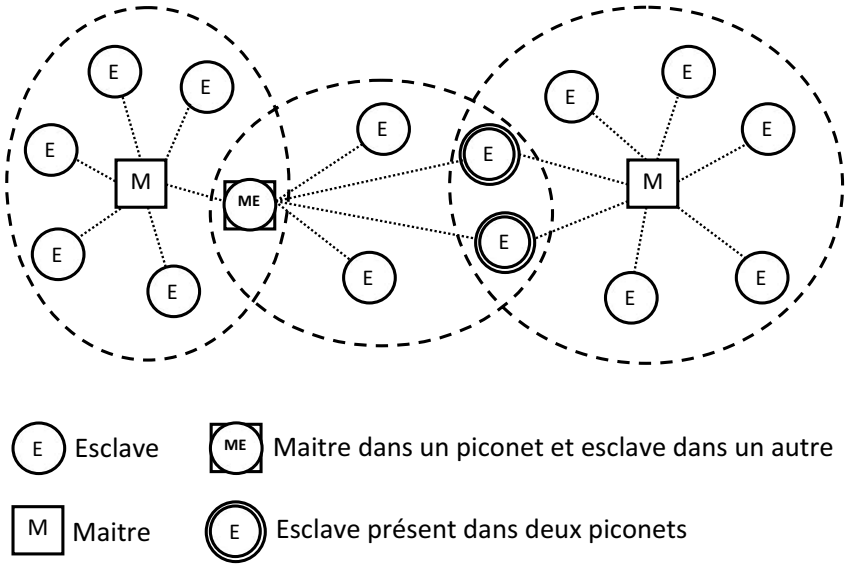

FIG. 1.1 – Scatternet formé de trois piconets

1.3 Applications des réseaux Bluetooth

Bluetooth a été créé pour la mise en place d'un réseau personnel sans fil, remplaçant, ainsi, le câblage. Nous examinons, dans ce qui suit, d'autres applications possibles des réseaux Bluetooth [63].

Réseaux ad hoc : Bluetooth peut être utilisé pour créer des connexions ad hoc entre utilisateurs permettant, ainsi, plusieurs applications telles que le partage de fichiers et les jeux en réseaux.

Station de base à la maison : des stations de base Bluetooth peuvent servir comme relais entre des équipements Bluetooth (généralement des téléphones mobiles) et des réseaux large bande tels que les réseaux téléphoniques commutés (RTC) et les réseaux informatiques. Pour le premier cas, un utilisateur à la maison peut utiliser son téléphone portable pour une communication qui passe par le RTC, minimisant, ainsi, le prix des communications. Pour le deuxième cas, un utilisateur peut utiliser son téléphone portable pour accéder aux réseaux informatiques et utiliser ainsi diverses applications tel que la voix et la vidéo.

Réseaux d'entreprise : un réseau Bluetooth d'entreprise peut être vu comme une extension du concept de "station de base" en une solution multi-cellules multi-sites qui peut offrir des solutions de communication mobile aux clients telles que la voix et la transmission de données. Cependant, l'implémentation de cette solution nécessite la résolution de plusieurs problèmes tels que les hand-over d'une station de base Bluetooth à une autre.

Commerce mobile : un réseau Bluetooth peut être établi entre un téléphone mobile et une machine vendeuse (distributeur) pour l'achat de bien tels que des produits alimentaires, des tickets de voyage ou de "parking". Le payement sera géré par la puce du téléphone. Cependant, des mécanismes de sécurité doivent être mis en place pour la sécurisation des communications.

Services d'information : des organisations touristiques et de transport ainsi que d'autres peuvent mettre en place des réseaux Bluetooth pour présenter, à leurs clients, des services d'informations tels que des cartes géographique, des tours en vidéo, l'histoire locale de régions, des listes d'adresses, etc.

Ainsi, la majorité des applications Bluetooth sont exigeantes en qualité de service vu qu'elles incluent la voix et la vidéo qui présentent des contraintes de débit et de temps.

1.4 Types de communication dans Bluetooth

Les communications dans un réseau Bluetooth peuvent être synchrones ou asynchrones, assurées moyennant trois types de lien : *Synchronous Connection Oriented* (SCO), *Extended Synchronous Connection Oriented* (eSCO) et *Asynchronous Connection Less*(ACL) :

Lien SCO : ce type de lien permet une connexion point à point symétrique entre

le maître et un esclave, permettant la communication dans des slots réservés. Les liens SCO sont, généralement, utilisés pour transporter des données avec contrainte de temps comme la voix. Le maître peut supporter jusqu'à trois connexions SCO avec un même esclave ou avec différents esclaves. L'esclave peut supporter jusqu'à trois connexions SCO avec le même maître ou uniquement deux liens SCO avec deux maîtres différents. Le maître envoie des paquets SCO à l'esclave après des intervalles de slots réguliers (Tsco). Un paquet SCO couvre un seul slot de temps réservé à cette connexion et s'appelle slot maître-à-esclave. Un esclave ayant une connexion SCO avec un maître, doit répondre avec un paquet SCO dans le slot esclave-à-maître qui suit le slot maître-à-esclave. Aucun mécanisme de retransmission, de détection et de correction d'erreur n'est prévu pour les paquets SCO.

Lien eSCO : ce type de lien a été introduit dans la version 1.2 des spécifications de Bluetooth. Comme pour le cas d'un lien SCO, le maître réserve, à un intervalle de temps régulier, des slots pour un esclave donné. Chaque esclave ne peut supporter qu'un seul lien eSCO. De plus, à la différence du lien SCO, un lien eSCO peut être symétrique ou asymétrique : à la création, le maître et l'esclave négocient le type de paquet et la périodicité des slots réservés, notée TeSCO. Contrairement aux liens SCO, un mécanisme de retransmission s'appuyant sur un CRC de 16 bits a été implémenté pour les liens eSCO. En cas d'erreur, l'esclave ou le maître peuvent immédiatement retransmettre les données erronées dans un intervalle de temps appelé fenêtre de retransmission de longueur WeSCO. Si la fin de l'intervalle est atteinte, le paquet est rejeté. L'application peut lier la taille de cet intervalle à l'échéance de remise des paquets du flux concerné.

Lien ACL : Dans les slots qui ne sont pas réservés pour les connexions SCO, le maître peut échanger des paquets avec n'importe quel esclave. On dit que le maître a établit une connexion ACL avec les esclaves. Cette dernière fournit une connexion à commutation de paquet entre le maître et tous les esclaves actifs participant au piconet. Entre un maître et un esclave, une seule connexion ACL peut exister. L'esclave répond par un paquet ACL dans un slot esclave-à-maître si et seulement s'il a été adressé par un paquet ACL dans le slot maître-à-esclave qui précède. Ainsi, si l'esclave n'arrive pas à décoder l'adresse esclave dans l'entête du paquet, il n'envoie pas un paquet ACL au maître. Un paquet ACL qui n'est pas adressé à un esclave particulier est considéré comme un paquet broadcast et peut être lu par n'importe quel esclave.

La figure 1.2 illustre un scénario de transmissions simultanées de trafics SCO et ACL. Le trafic SCO est transmis périodiquement dans des slots réservés. Le trafic ACL est transmis entre les transmissions SCO.

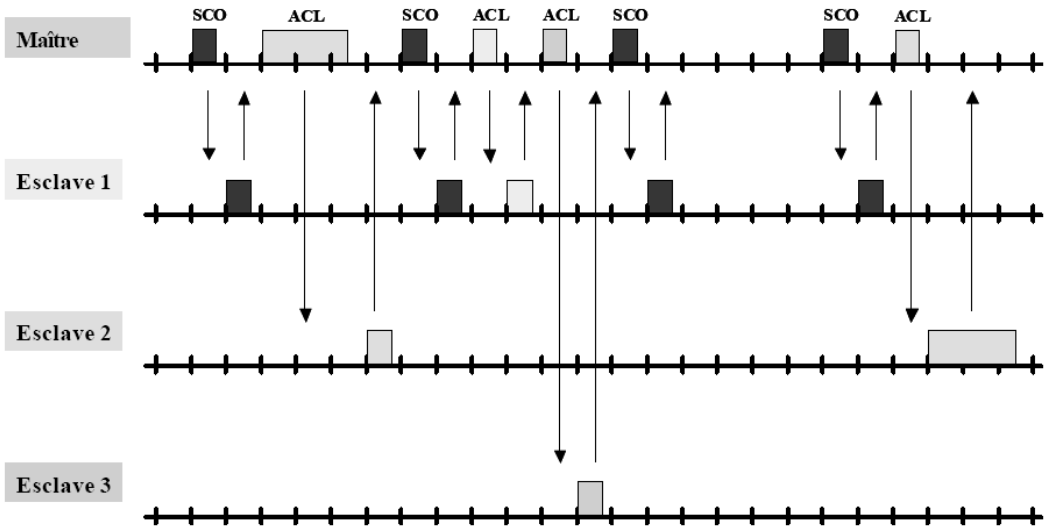

FIG. 1.2 – Exemple de transmissions simultanées de trafics SCO et ACL

1.5 Pile protocolaire de Bluetooth

La pile de protocoles de Bluetooth est illustrée par la figure 1.3. Les trois premières couches (Physique, Baseband et Link Manager) sont généralement regroupées et implémentées, sous forme matérielle, sur un unique système appelé Contrôleur Bluetooth. La couche L2CAP et l'application constituent la partie hôte du système. Elles sont généralement implémentées, sous forme logicielle, sur l'hôte. Nous présentons, dans ce qui suit, le fonctionnement des couches physique, baseband, link manager et L2CAP.

1.5.1 Couche physique (RF : Radio Frequency)

Cette couche s'occupe de l'émission et de la réception des ondes radio. Elle définit les caractéristiques de la transmission telles que la bande de fréquence, les canaux, la modulation, etc.

La transmission se fait dans la bande de fréquence 2.4 GHz ISM (Industrial, Science and Medecine). Dans la plupart des pays du monde, cette bande de fréquence varie entre 2400 et 2483,5 MHz. Cependant, quelques pays présentent des limitations nationales influençant la largeur de cette bande. En France, par exemple, les fréquences varient entre 2446,5 et 2483,5 MHz. Cette bande est divisée en canaux RF de 1MHz. Ce qui fait 79 canaux de 1MHz, numérotés de 0 à 78, commençant par 2402 MHz (f = 2402 + k MHz, k=0,..., 78). Une bande de garde est mise en place aux extrémités inférieure et supérieure de la bande. Pour la bande de fréquences limitée, il y a uniquement 23 canaux RF (f = 2454 + k MHz, k =0,...,22).

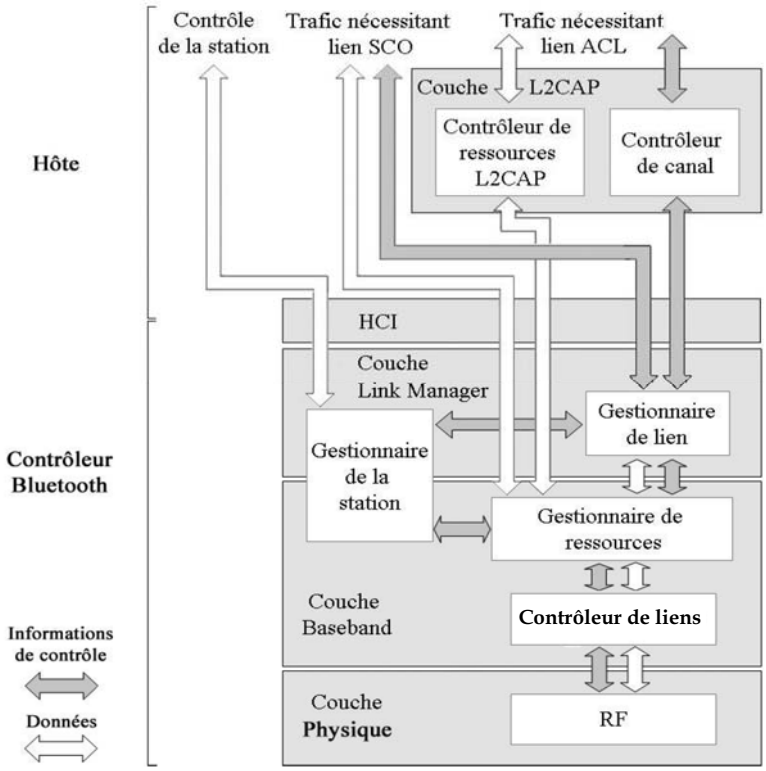

FIG. 1.3 – Pile protocolaire de Bluetooth

1.5.2 Couche baseband

La couche baseband définit les adresses matérielles BD_ADDR (Bluetooth Device Adress) des périphériques Bluetooth et elle assure les fonctionnalités de base pour la transmission de données telles que la synchronisation, la transmission des paquets, le contrôle d'erreurs et l'accès au médium. Ces fonctionnalités sont implémentées au niveau du gestionnaire de ressources et du contrôleur de liens.

1.5.2.1 Contrôleur de liens

Ce module assure trois fonctions de base :
- Encapsulation des données dans les paquets Baseband.
- Contrôle et paramétrage du lien physique (fréquences utilisées, etc).
- Gestion de la signalisation utilisée dans les paquets Baseband et ce, en collaboration avec le gestionnaire de ressources.

1.5.2.2 Gestionnaire de ressources

Ce module, équivalent à un ordonnanceur, gère l'accès au médium entre les paquets Baseband en attente de transmission pour les différents liens SCO, eSCO et ACL établis entre le maître et un esclave. Ainsi ce module assure :
- **L'ordonnancement local** des paquets Baseband pour chaque lien (SCO, eSCO, ACL)

– **L'ordonnancement intra-piconet** vu qu'il assure le polling des esclaves sur un piconet. Pour une transmission de données "Full Duplex" entre maître et esclave, Bluetooth implémente un mécanisme d'accès de type Time Division Duplex (TDD). Le maître doit commencer sa transmission dans un slot de temps paire et l'esclave la commence dans un slot de temps impaire (voir figure 1.4). Le maître du piconet a la charge du "polling" des esclaves. Un esclave n'est autorisé à transmettre dans un slot impair, que s'il a été scruté (pollée) dans le slot pair précédent. De plus, toute communication directe entre esclaves est impossible. Les paquets Bluetooth sont de longueur 1, 3 ou 5 slots. Si, en scrutant un esclave, le maître n'a rien à lui transmettre, il émet un paquet Poll qui ne contient pas de données utilisateur. Respectivement, si l'esclave n'a rien à transmettre, il émet un paquet Null. Les paquets Poll et Null sont de taille 1 slot.

Nous notons, donc, l'importance de ce module pour le support de la QoS vu qu'il est le responsable de l'ordonnancement des paquets qui peut être effectué en tenant compte des contraintes de QoS requises par l'application pour chaque type de lien.

Ce module gère aussi l'allocation des ressources radio en tenant compte des négociations effectuées avec les modules qui lui sont connectés. Ainsi, il permet la sélection du type de paquet utilisé (Les paquets Bluetooth sont de longueur 1, 3 ou 5 slots), la procédure d'acquittement des paquets et le contrôle de flux.

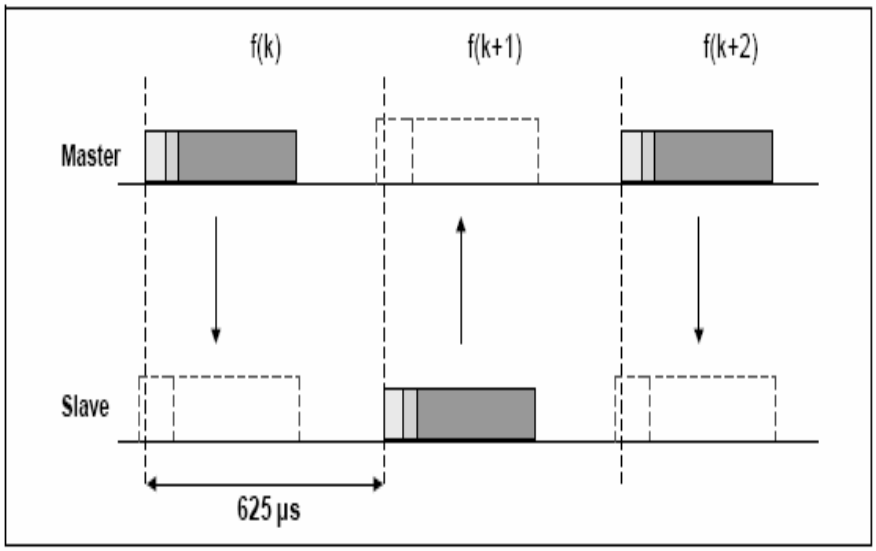

FIG. 1.4 – Slots de temps avec TDD

1.5.3 Couche Link Manager

Le protocole LMP (Link Manager Protocol) est utilisé pour contrôler et négocier tous les aspects du fonctionnement de la connexion Bluetooth entre deux périphériques, notamment la configuration et le contrôle de transports et de liaisons logiques, ainsi que pour contrôler les liaisons physiques. Par ailleurs, il est utilisé pour communiquer entre les gestionnaires des liaisons des deux périphériques qui sont connectés par le lien logique ACL.

Le protocole LMP est aussi responsable de la recherche de stations dans le voisinage, la gestion des modes d'économie d'énergie, la création, la modification et la libération des liens SCO, eSCO et ACL supportés par la station.

1.5.4 Couche Logical Link Control and Adaptation Layer Protocol (L2CAP)

La couche L2CAP prend en charge le multiplexage de protocoles de haut niveau, la segmentation et le réassemblage de paquets ainsi que la transmission de la qualité de service des informations.

Cette couche permet à d'autres protocoles et applications de plus haut niveau de transmettre et de recevoir des paquets de données des couches supérieures (SDU du protocole L2CAP) d'une longueur pouvant aller jusqu'à 64 Kbits. La couche L2CAP facilite également le contrôle de flux par canal et la retransmission via leur mode respectif.

La couche L2CAP fournit des canaux logiques, appelés canaux L2CAP, qui sont mappés sur des liaisons logiques L2CAP prises en charge par un lien logique ACL.

1.6 Économie d'énergie

L'économie d'énergie dans Bluetooth est assurée en implémentant les modes d'activité : actif, sniff, hold et park. Le mode actif est utilisé pour communiquer. Les modes sniff, hold et park sont des modes d'économie d'énergie. Nous expliquons davantage les différents modes dans les points suivants :

Mode actif : dans ce mode, l'équipement Bluetooth participe activement au canal. Le maître ordonnance les transmissions en se basant sur les demandes de trafic de et vers les esclaves. De plus, il supporte des transmissions régulières pour maintenir les esclaves synchronisés au canal. Les esclaves actifs se mettent à l'écoute du canal pendant les slots maître-à-esclave pour répondre aux messages reçus.

Mode sniff : Dans ce mode, les cycles d'activité pour l'écoute des esclaves peuvent être réduits. Si l'esclave participe à une connexion ACL, il doit écouter le trafic du maître à chaque slot de temps ACL. Avec le mode sniff, le nombre de slots de temps dans lesquels le maître peut commencer à transmettre à un esclave spé-

cifique est réduit. En effet, le maître peut commencer à transmettre uniquement dans des slots de temps spécifiques. Ces slots sont dits slots sniff et sont espacés régulièrement d'un intervalle Tsniff.

Mode hold : dans ce mode, l'esclave arrête temporairement le support des paquets ACL dans le canal. Les liens SCO peuvent continuer à être supportés.

Mode park : quand l'esclave n'a pas besoin de participer au canal du piconet mais qu'il veut rester synchronisé au canal, il peut entrer en mode park qui est un mode à faible consommation d'énergie. Dans ce mode, l'esclave n'utilise plus son adresse AM_ADDR (Active Membre Address) mais il reçoit deux autres nouvelles adresses à utiliser en mode park. Ces adresses sont PM_ADDR (Parked Member Address) et AR_ADDR (Access Request Address). La PM_ADDR est utilisée dans le cas où le maître veut déparquer l'esclave alors que la AR_ADDR est utilisée par l'esclave quand il demande d'être déparqué.

1.7 Ordonnancement pour le support de qualité de service dans Bluetooth

Dans cette section, nous décrivons, dans une première étape, les types d'ordonnancement dans les réseaux Bluetooth ainsi que les limites de l'ordonnancement Bluetooth. Dans une seconde étape, nous présentons un état de l'art des solutions d'ordonnancement dans un réseau Bluetooth en mettant en évidence leurs avantages et inconvénients.

1.7.1 Types d'ordonnancement dans Bluetooth

Les mécanismes d'ordonnancement des messages Bluetooth peuvent être classés en trois catégories [7] : ordonnancement local, ordonnancement intra-piconet et ordonnancement inter-piconet.

1. **Ordonnancement local :** c'est l'ensemble des règles utilisées par un équipement Bluetooth pour sélectionner un message de sa file d'attente. Au niveau de l'esclave, la file d'attente locale est notée $S \rightarrow M$ et au niveau du maître, elle est notée $M \rightarrow S$ (une file d'attente par esclave).
2. **Ordonnancement intra-piconet :** il s'agit des méthodes de polling formant le Medium Access Control (MAC) pour les réseaux Bluetooth. En effet, le polling est l'ensemble des règles qui déterminent quand est ce que le maître du piconet change d'un esclave à un autre et quel esclave choisir.
3. **Ordonnancement inter-piconet :** c'est le mécanisme qui détermine le piconet dans lequel une passerelle Bluetooth doit être présente à un moment donné dans le scatternet.

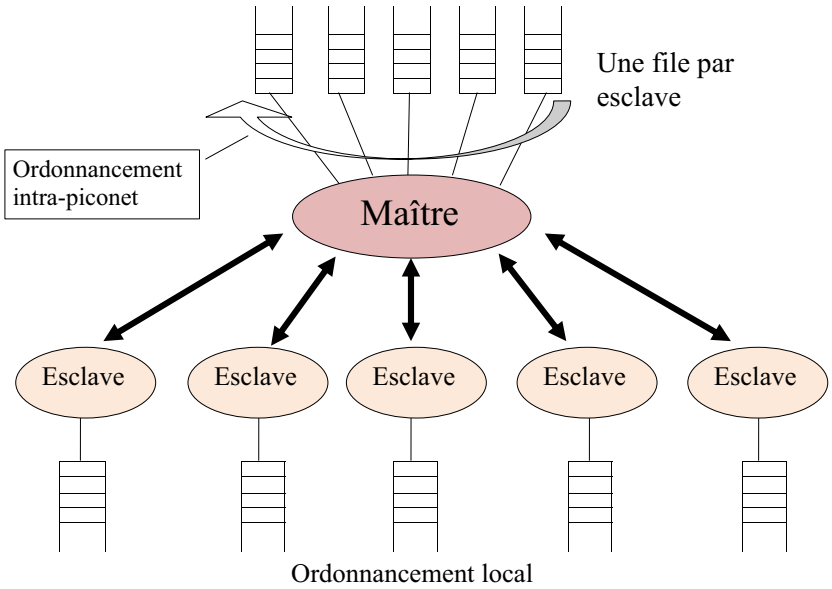

FIG. 1.5 – Ordonnancement local et intra-piconet pour un piconet de 5 esclaves

Dans nos travaux, nous nous intéressons à l'ordonnancement local et intra-piconet (voir figure 1.5) afin de garantir une qualité de service en tenant compte des contraintes de degré d'importance et d'échéance de remise des messages.

La combinaison de l'ordonnancement local et intra-piconet forme ce qu'on appelle ordonnancement global. Nous nous intéressons, donc, à ce niveau d'ordonnancement.

Comme déjà mentionné, le piconet Bluetooth est formé d'un maître gérant au maximum 7 esclaves actifs. Le maître maintient au moins une file d'attente par esclave, dite file maître-à-esclave et notée $M \rightarrow S$. L'esclave maintient au moins une file d'attente esclave-à-maître notée $S \rightarrow M$.

L'ordonnancement Bluetooth doit satisfaire certaines propriétés telles que l'efficacité et l'équité. En effet, les ressources des équipements Bluetooth sont limitées et doivent, donc, être utilisées d'une manière efficace. En d'autres termes, le rapport entre le nombre des paquets POLL et NULL et le nombre de paquets de données doit être minimisé. De plus, le poller (maître) doit gérer les esclaves d'une manière équitable par rapport à une métrique donnée telle que le débit, le temps d'attente et le temps de service [20]. Par exemple, pour la métrique débit, le poller doit allouer le même débit pour tous les esclaves.

1.7.2 Limites de l'ordonnancement Bluetooth pour le support de la qualité de service

Pour l'ordonnancement local des messages, Bluetooth utilise deux files d'attente. La première file d'attente est implémentée au niveau de la couche Baseband et est utilisée pour la transmission des paquets (déjà fragmentés) sur le support.

La deuxième file d'attente est implémentée au niveau de la couche L2CAP et gère les paquets avant qu'ils soient fragmentés et envoyés à la couche Baseband.

Au niveau de la couche baseband, les spécifications Bluetooth recommandent l'utilisation d'une file d'attente FIFO. Cependant, FIFO ne tient pas compte des contraintes degré d'importance et échéances des messages, ce qui ne permet pas une différenciation de service entre les différents messages. De plus, pour les liens ACL, l'ordonnanceur de type FIFO transmet tous les fragments d'un même paquet avant de passer aux fragments d'un second paquet et ce, parce qu'il n'y a pas un moyen de repérer les fragments d'un même paquet. De même, pour les liens eSCO et SCO, l'ordonnanceur transmet les messages dans leur ordre d'arrivé.

Au niveau de la couche L2CAP, les spécifications Bluetooth ne précisent aucun mécanisme d'ordonnancement des messages reçu à partir de l'application. Pour l'ordonnancement intra-piconet, les spécifications Bluetooth recommandent l'utilisation d'un ordonnanceur de type one round robin (1-RR). Cependant, 1-RR ne permet pas une différenciation de service et ne peut, donc, pas garantir une QoS pour un flux donné. Pour l'ordonnancement inter-piconet, les spécifications de Bluetooth ne proposent aucun type d'ordonnanceur.

En conclusion, les types d'ordonnancement recommandés par les spécifications de Bluetooth ne permettent pas une différenciation de service et ne présentent, donc, pas de garanties de QoS.

1.7.3 État de l'art des algorithmes de segmentation des messages

La segmentation des messages applicatifs est effectuée au niveau de la couche L2CAP. Les fragments générés sont de taille 1, 3 ou 5 slots. L'algorithme de segmentation permet la sélection des tailles des fragments (1, 3 ou 5 slots) de telle manière à optimiser l'utilisation des ressources radio. Cette sélection doit minimiser, d'une part, l'overhead (entêtes des paquets) en favorisant, par exemple, l'utilisation de paquets de 5 slots et d'autre part le nombre d'octets perdu lorsque le champ de données n'est pas entièrement rempli.

Les spécifications de Bluetooth n'indiquent pas d'algorithme de segmentation. Ainsi, des algorithmes ont été proposés dans la littérature :

SAR Intelligent (SAR-I) : cet algorithme, décrit dans [41], vise à minimiser la perte des lots. Il permet d'adapter les tailles des paquets des files $M \rightarrow S$ et $S \rightarrow M$ selon le taux d'arrivée à chaque file. Initialement, les deux files ont une taille de paquet égale à 1 slot. Cette taille reste intacte tant que les files $M \rightarrow S$ et $S \rightarrow M$ présentent des taux d'arrivée égaux. Dans le cas contraire, la taille des paquets de la file ayant le plus grand taux d'arrivée devient supérieure à celle des paquets de la deuxième file. Ainsi, l'algorithme SAR-I permet de réduire la perte de slots. Cependant, l'esclave nécessite la connaissance du taux d'arrivé au niveau du maître et vice versa. À cet effet, les auteurs de [41] proposent l'utilisation d'un bit d'information qui indique si le taux d'arrivée est élevé ou faible. Cependant, ce

bit ne permet pas de comparer quantitativement les taux d'arrivée. Par exemple, si le maître signale à l'esclave qu'il a un taux d'arrivé élevé et vice versa, lequel sera favorisé ?

SAR-Best-Fit (SAR-BF) : cet algorithme, décrit dans [42], vise à minimiser le nombre de slots nécessaires pour la transmission du message à transmettre. Il permet de répartir les octets d'un message sur des paquets en commençant par les paquets de taille 5 puis 3 puis 1 slot. Par exemple, un message de 895 octets est divisé en 2 paquets de taille 5 slots (339 octets), un paquet de taille 3 slots (183 octets) et deux paquets de taille 1 slot (un de 27 octets, un de 7 octets).

SAR-Optimum Slot Utilization (SAR-OSU) : cet algorithme, décrit dans [42], vise à minimiser le nombre de fragments (paquets). Ainsi, il divise le message applicatif sur des paquets de taille 5 slots et met les octets restant sur un paquet de taille juste supérieure à la taille nécessaire. Par exemple, un message de 523 octets est divisé en deux paquets de taille 5 slots (un de 339 octets, un de 184 octets) et un message de 522 octets est divisé en un paquet de taille 5 slots (339 octets) et un paquet de taille 3 slots (183 octets).

SAR-Channel State Dependent (SAR-CSD) : cet algorithme, décrit dans [44], vise à maximiser l'utilisation du lien en utilisant des fragments (paquets) multi-slots. Son principe de fonctionnement est équivalent à celui de (SAR-OSU). Cependant, SAR-CSD introduit une première phase qui consiste à sélectionner le type de paquet (DH ou DM) pour la fragmentation des messages spécifique à un esclave donnée et ce, en se basant sur un tableau de fréquences. Ce tableau indique l'état des fréquences entre le maître et tous les esclaves du piconet (Better, Good, Bad). Le type DH est sélectionné si le nombre de fréquences "better" et "good" divisé par le nombre total de fréquence est supérieur à un seuil particulier. Dans le cas contraire le type DM est sélectionné.

Pattern Matching Polling (PMP) : cet algorithme, décrit dans [43],vise à maximiser le remplissage du champ de données d'un paquet (fragment) afin de minimiser le nombre d'octets perdus. Il permet aussi de déterminer la taille du fragment (paquet) et l'instant de polling d'un esclave en fonction de la loi d'arrivée des messages dans la file $M \rightarrow S$ ou $S \rightarrow M$ concernée.

Dans la suite de cette thèse, nous optons pour l'algorithme de segmentation SAR-BF qui vise à minimiser le nombre de slots nécessaires pour transmettre un message.

1.7.4 État de l'art de l'ordonnancement local

Concernant l'ordonnancement local, les spécifications Bluetooth [10] recommandent l'utilisation des files d'attente FIFO pour la transmission des messages. Ces derniers sont alors transmis par ordre croissant de leurs temps d'arrivée. Un

tel type d'ordonnancement ne permet pas de prendre en compte les contraintes de degré d'importance et d'échéances.

Dans la littérature, peu de travaux ont proposés des améliorations de la gestion des files d'attente locales pour la prise en compte des échéances et des priorités. De plus, ces travaux, que nous présentons dans les points suivant présentent des limites.

Priority Queuing at Master (PQM) : Ce mécanisme [12] concerne essentiellement les configurations asymétriques où le trafic du maître vers les esclaves est favorisée. Le maître prend en compte deux classes de paquets. Localement, il maintient pour la classe la plus prioritaire une file d'attente FIFO pour tous les esclaves. Pour la classe la moins prioritaire, il maintient une file FIFO par esclave. Aucun mécanisme n'est spécifié pour l'ordonnancement local des files d'attente des esclaves.

1-RR amélioré : Il s'agit [7, 9] d'une amélioration du mécanisme d'ordonnancement Bluetooth. Localement, les paquets sont classifiés en des différentes classes. La classe la plus prioritaire est servie en premier. Au niveau de chaque classe, l'ordonnancement EDF (Earliest Deadline First) est utilisé. Cette méthode tient compte, localement, des deux contraintes qui nous intéressent : priorité et échéance des messages. Néanmoins, elle a été étudiée en établissant une borne déterministe en considérant le pire cas. Mais, elle reste pessimiste puisque la situation pire cas se produit rarement.

1.7.5 Etat de l'art de l'ordonnancement intra-piconet (Polling)

L'ordonnancement intra-piconet est géré par le maître et est souvent assuré par un système de polling. Il représente l'ensemble des règles qui déterminent quel esclave doit être pollé par le maître.

Plusieurs algorithmes d'ordonnancement intra-piconet ont été proposés dans la littérature. Ces algorithmes diffèrent principalement par les règles qui déterminent quel esclave va être pollé, ce qui affecte les performances du système. Nous présentons, dans une première étape, le système de polling dans Bluetooth vu qu'il diffère des systèmes de polling classique. Dans une seconde étape, nous décrivons les solutions d'ordonnancement intra-piconet présentées dans la littérature. De même, nous comparons ces solutions selon des critères que nous définissons.

1.7.5.1 Système de polling dans Bluetooth

Un système de polling est un système de files d'attente dans lequel un serveur visite un ensemble de files pour servir les clients [13] selon une méthode de polling. Une méthode de polling est un ensemble de règles qui déterminent quand est ce que le serveur bascule d'une file à une autre, quelle est la file suivante à visiter par le serveur, le nombre de clients de la file qui vont être servis, etc.

Ce modèle de système de polling a été étudié dans la littérature et appliqué à plusieurs problèmes de réseaux de communication. Il peut aussi être adopté pour décrire le fonctionnement MAC au niveau d'un piconet du réseau Bluetooth. Cependant, le modèle de polling dans le fonctionnement de Bluetooth diffère des modèles classiques analysés dans la littérature vu qu'il doit prendre en compte les points suivants :

- Toutes les communications sont bidirectionnelles ; il ne peut pas exister un paquet descendant (du maître vers un esclave) sans un paquet ascendant (d'un esclave vers le maître) et vice versa.
- Après la transmission d'un paquet du maître à l'esclave, ce dernier peut envoyer un paquet au maître. Ainsi, les visites des files $M \rightarrow S$ et $S \rightarrow M$ effectuées par le serveur sont intimement corrélées. En effet, une transmission $M \rightarrow S$ est immédiatement suivie d'une transmission $S \rightarrow M$.
- Bien que le maître gère la procédure d'ordonnancement MAC (polling), il n'a qu'une connaissance partielle des états des files : il connaît les états de toutes les files $M \rightarrow S$ mais il ne peut pas avoir une information à jour sur les états des files $S \rightarrow M$, même si des messages de signalisation explicites lui sont communiquées à partir des esclaves.
- Toute communication entre deux esclaves doit être routée par maître. Il ne peut pas y avoir une communication directe entre les esclaves.

1.7.5.2 Polling basé sur la méthode Round Robin

Round Robin (RR) est l'un des algorithmes d'ordonnancement les plus simples et les plus utilisés. L'application de cette méthode pour l'ordonnancement intra-piconet signifie que les esclaves sont scrutés dans un ordre cyclique. Différentes approches basées sur RR ont été proposées dans la littérature [13]. Ces approches diffèrent par le critère utilisé pour la définition du cycle et le temps accordé à chaque paire maître-esclave pour leur communication :

Pure Round Robin (PRR) : C'est l'algorithme de polling proposé par la spécification de Bluetooth [10]. Dans cet algorithme, appelé aussi, One Round Robin (1-RR), un ordre cyclique fixe est défini et une seule chance pour transmettre est donnée à chaque paire de files maître-esclave selon l'ordre cyclique : a chaque cycle, pour chaque esclave, un seul paquet du maître et un seul paquet de l'esclave sont servis. Ainsi, cette méthode n'est pas exhaustive dans le sens qu'elle peut passer à une nouvelle paire de files sans vider la paire de files courante.
Cet algorithme partage le nombre total de scrutin (polls) entre les esclaves sans tenir compte des différents besoins de chacun d'entre eux. Par conséquent, quelques esclaves sont scrutés sans nécessité tandis que d'autres esclaves à grande charge de trafic sont scrutés moins qu'il le faut. PRR est intéressant seulement dans le cas où le trafic global est faible ou les débits sont similaires. Dans n'importe quelle autre situation, il est très inefficace [14].

Exhaustive Round Robin (ERR) : comme PRR, un ordre fixe est défini mais

la méthode est exhaustive. Le maître continue à poller le même esclave tant que les files du maître et de l'esclave considéré ne sont pas vides. Il ne passe au prochaine paire de files maître-esclave que lorsque les files du maître et de l'esclave courant sont vidées. Ainsi, un esclave peut consommer toute la bande passante s'il a toujours des paquets à transmettre. En conséquence, le problème de famine peut se manifester et le polling est ainsi considéré non déterministe.

Exhaustive Pseudo-cyclic Master queue length (EPM) : un ordre cyclique dynamique est défini au début de chaque cycle suivant un ordre décroissant des longueurs des files $M \rightarrow S$ (chaque paire maître-esclave est visité exactement une fois par cycle). Le maître polle le même esclave tant que la file $M \rightarrow S$ qu'il maintient n'est pas vide. Les performances de cet algorithme se dégradent lorsque le trafic $S \rightarrow M$ est plus important que le trafic $M \rightarrow S$.
Les auteurs de [13] ont montré que :

- PRR est le moins performant des trois algorithmes vu qu'il donne le plus haut temps de réponse.
- ERR et EPM ont pratiquement les mêmes performances. Ainsi, EPM n'est pas intéressant parce qu'il nécessite, contrairement à ERR, une connaissance partielle des états des files d'attentes.
- ERR est efficace dans le cas de scénarii de charge symétrique.
- Dans ERR, les esclaves qui génèrent un trafic plus grand que la capacité du système, peuvent monopoliser le canal, ce qui peut causer une distribution inéquitable de la bande passante.

Pour résoudre ce dernier problème, deux autres méthodes d'ordonnancement intrapiconet ont été proposées dans [13] que nous présentons dans ce qui suit :

Limited Round Robin (LRR) : Cette méthode résout le problème d'iniquité d'ERR en limitant le nombre de transmissions que chaque paire peut assurer par cycle. Cependant, les esclaves qui n'ont rien à transmettre sont pollés, entraînant par conséquent un gaspillage de la bande passante.

Limited and Weighted Round Robin (LWRR) : cette méthode est une amélioration de LRR dans le sens qu'elle minimise le taux de visite des files qui ont été trouvé vides lors du dernier poll. Pour ce faire, elle attribue des poids dynamiques aux paires maître-esclave. Initialement, à chaque esclave est affecté un poids MP (Maximum Priority). Ensuite, chaque esclave est scruté selon LRR. S'il apparaît un échange de paquets POLL-NULL, le poids de l'esclave correspondant est décrémenté de 1. Si ce poids atteint la valeur 1, l'esclave doit attendre $MP - 1$ cycles pour être scruté de nouveau. Lors d'un échange de données maître-esclave le poids de cet esclave est incrémenté jusqu'à la valeur MP.
Ainsi, cette méthode minimise le polling des esclaves qui n'ont rien à transmettre. Cependant, elle présente deux limites principales dues à la variation du nombre de slots dans un cycle de polling :

- Si les cycles de polling précédents possèdent un grand nombre de slots, un esclave inactif doit attendre longtemps ($MP-1$ cycles) pour avoir une chance pour transmettre.
- Si les cycles de polling précédents possèdent un nombre réduit de slots, un esclave inactif est fréquemment pollé. Ceci peut dégrader les performances du système.

1.7.5.3 Polling améliorant la méthode Round Robin

Des méthodes d'ordonnancement intra-piconet ont été proposées afin d'améliorer les performances des méthodes d'ordonnancement basées sur RR [16, 17] :

Pseudo-Random Cyclic Limited slot-Weighted Round Robin (PLsWRR) : cette méthode [16] améliore LWRR dans le sens que l'ordre de polling est déterminé d'une manière pseudo-aléatoire et que les poids des slots sont changés selon l'activité de l'esclave dans le cycle précédent. De plus, PLsWRR utilise le nombre de slots pour réduire le polling des esclaves inactifs alors que LWRR utilise le nombre de cycles pour le même but. Initialement, à chaque esclave est assigné un poids de slot égale à "Max-Slot Priority" (MSP). Quand un esclave est pollé mais il n'y a pas transfert de données, le poids de slot de ce dernier est décrémenté du nombre de slots du cycle précédent. Comme dans LWRR, le poids minimum est 1 et s'il y a échange de données entre l'esclave et le maître, le poids de slot de l'esclave est augmenté jusqu'à MSP. Ainsi, PLsWRR garantie que les esclaves attendent au maximum MSP slots pour avoir une chance d'être pollé. Ceci est plus fiable que pour le cas de LWRR où un esclave doit attendre $MP-1$ cycles dont la longueur est variable. En conséquence, PLsWRR résout les deux problèmes de LWRR.

Deficit Round Robin (DRR) : cette méthode cherche principalement l'équité entre les esclaves. Elle traite les files d'attente de point de vu taille des données et non pas nombre de messages. Son principe est le suivant [17] :

- Chaque file d'attente admet un compteur (dc) initialisé à 0
- L'ordonnanceur essai de servir un quantum q de donnée à partir d'une file non vide.
- Le paquet à l'entête de taille T est servi si $T \leq q + dc$
- Si le paquet est servi alors $dc \leftarrow q + dc - T$, sinon $dc \leftarrow q + dc$

Round-Robin with FCFS Preemption (RR-FCFS) : cette méthode [55] ne tient compte que de l'état des files d'attente du maître qui sont vue comme une seule file d'attente FCFS. Quand cette file est vide, le maître polle les esclaves selon 1-RR. Dès que cette file devient non vide, le maître arrête le polling selon 1-RR et commence le polling selon la discipline FCFS jusqu'à ce que la file devienne vide. En ce moment, il reprend le poling 1-RR. Les auteurs ont montré, par simulation, que les performances de RR-FCFS sont comparables à ceux de ERR et 1-RR et aucun des trois méthodes n'est définitivement meilleure que les autres pour tout type de trafic (symétrique, asymétrique, uniforme, etc.).

Look Ahead (LA) : Dans cette méthode [18], le maître polle les esclaves selon l'algorithme suivant :

1. Il détermine la taille du paquet en tête (HOL) de chaque file $M \rightarrow S$ combiné avec $S \rightarrow M$, notée CMSQ (Combined Master Slave Queue).
2. Il assigne à ces paquets des priorités selon leurs tailles (le paquet qui a la plus grande taille aura la plus haute priorité).
3. Il sert le paquet de plus haute priorité qui peut tenir sur les slots disponibles.
4. Si aucun paquet HOL ne répond aux contraintes de l'étape 3, le maître attend le début des prochains slots de temps disponibles et sert le paquet HOL le plus prioritaire.
5. répéter les étapes 1 à 4.

Cet algorithme permet une différenciation de service, cependant il présente trois limites principales :

1. à chaque fois que le maître sert un paquet, il doit déterminer le paquet le plus prioritaire suivant, ce qui résulte en un overhead de calcul important (O(logN) [18], avec N le nombre d'esclave actif).
2. le maître peut servir des esclaves au détriment d'autres, ce qui peut résulter en un problème de famine.
3. le maître se base sur des files $CMSQ_i$ où chacune représente la combinaison de la file $M \rightarrow S_i$ avec $S_i \rightarrow M$ de l'esclave i. Cependant, le maître ne peut pas avoir une information exacte du paquet HOL de la file $S_i \rightarrow M$.

Round Robin with Look Ahead (LARR) : Cet algorithme [18] est une combinaison de LA et 1-RR. Il a été développé pour résoudre les problèmes de LA. En effet, LARR ne détermine plus le paquet HOL le plus prioritaire des toutes les files $M \rightarrow S$ mais il sert les esclaves selon 1-RR. La seule différence avec 1-RR est que si les slots disponibles ne sont pas suffisant pour servir le paquet HOL de l'esclave courant, LARR tente de servir le paquet HOL de l'esclave suivant (selon 1-RR). Si aucun paquet HOL ne peut être servi, le maître attend le début des prochains slots de temps disponibles et sert le paquet HOL de l'esclave courant. Bien que LARR présentent des solutions au problème d'overhead et de famine de LA, il ne permet pas d'avoir une information exacte du paquet HOL de la file $S_i \rightarrow M$. Les auteurs proposent d'utiliser le champ réservé du paquet récemment transmis de l'esclave vers le maître. Cependant, entre temps, le paquet HOL de la file $S_i \rightarrow M$ peut changer.

1.7.5.4 Polling adaptatifs

Les solutions d'ordonnancement intra-piconet adaptatifs cherchent à s'adapter aux trafics générés afin de répondre aux besoins d'équité, de délai et d'efficacité.

Fair Exhaustive Polling (FEP) : Cette méthode [22] vise principalement l'équité

et l'efficacité. Elle divise les esclaves en deux groupes : esclaves actifs et esclaves inactifs. Ensuite, elle scrute seulement les esclaves actifs d'une manière Round Robin. Pour mesurer l'activité de l'esclave, cette méthode utilise le nombre des polls successifs inutilisés ou le taux de succès moyen des polls. Si l'esclave est considéré inactif, il devient membre du groupe inactif. Chaque esclave i peut définir un intervalle maximum d'inter-poll (Tpolli). Cet intervalle est utilisé par l'ordonnanceur FEP pour scruter un esclave inactif régulièrement afin de tester s'il est devenu actif ou non. Si les demandes de trafic sont connues à l'avance, l'ordonnanceur FEP peut être utilisé pour partager la bande passante entre les esclaves d'une manière plus efficace que PRR.

Predictive Fair Poller (PFP) : comme FEP, cette méthode [19–21] vise l'efficacité et l'équité. Le maître prévoit pour chaque esclave si des données existent ou non en se basant sur le résultat du dernier POLL et des taux d'arrivée exprimés par les esclaves. Ensuite, il polle l'esclave possédant la plus haute priorité de point de vu fraction de bande passante déjà accordée à l'esclave et existence de donnée à transmettre.

Efficient double cycle scheduling (EDC) : cette méthode de polling [**?**,38,39] vise essentiellement l'efficacité. Elle définit deux cycles de polling correspondant à l'ordonnancement "uplink" et l'ordonnancement "downlink". Durant ces deux cycles, le maître met à jour la liste E(DW) des esclaves à scruter en "downlink" et celle E(UP) des esclaves à scruter en "uplink" et ce, en se basant sur des informations sur les charges des trafics des cycles précédents. E(DW) est déterminé par le maître, en se basant sur la longueur des files d'attente. Pour E(UP), si un esclave a envoyé un paquet avec un payload nul au cours du cycle précédent, sa fenêtre de poling est doublée (jusqu'à un certain seuil). Sinon elle est remise à 1 et l'esclave est scruté dans le cycle suivant.

Flow adapted polling Ce mécanisme de polling vise essentiellement l'efficacité en réduisant la fréquence de polling des esclaves à faible volume de données à transmettre. Dans [42] deux algorithmes de ce type sont présentés :

1. **Adaptive Flow based Polling (AFP)** : cet algorithme met à jour la fréquence de polling d'un esclave (initialement fixé à une valeur par défaut) à chaque transmission d'un paquet. Il la remet à la valeur par défaut si l'esclave correspondant a un nombre de paquets npq à transmettre supérieur à un certain seuil $seuil_{npq}$. Il la laisse intacte si $npq < seuil_{npq}$. Il la divise par deux si l'esclave répond avec un paquet NULL à un paquet POLL.

2. **Sticky Adaptive Flow based Polling (SAFP)** : SAFP est une extension d'AFP dans le sens qu'il inclut une option pour réduire le délai d'attente moyen dans les files. Les esclaves sont scrutés selon RR. Si un esclave possède un volume de données supérieur à un certain seuil, il est autorisé à transmettre un nombre maximum de paquets sinon il transmet un seul paquet.

Priority scheduling and exponential back-off mechanism : cette méthode de polling [40] est basé sur les priorités qui sont relié aux délais tolérables des paquets. Les esclaves sont divisé en deux groupes : esclaves "real time" et esclaves "best effort". A chaque esclave "real time" est assigné une priorité unique basé sur son délai tolérable maximum. En cas d'égalité, les requêtes qui arrivent en premier sont considérées plus prioritaires. Les esclaves "best effort" sont assignés la même priorité qui est la plus basse de toutes les priorités considérées. L'algorithme d'ordonnancement classe les esclaves "real time" selon leurs priorités et traite les esclaves "best effort" selon la méthode round robin. Ainsi, le maître délivre, le plutôt possible, les messages qu'il reçoit à partir des esclaves "real time" alors qu'il peut retarder les paquets reçus à partir des esclaves "best effort". Cette méthode de polling propose aussi une minimisation du polling des esclaves inactifs. Elle utilise un compteur pour chaque esclave "real time". Ce compteur est remis à zéro quand l'esclave correpondant répond avec un paquet NULL. En ce moment, le maître considère que l'esclave courant n'a plus de paquets à transmettre et passe à l'esclave suivant (de priorité juste inférieur). Pour les esclaves "best effort", l'algorithme d'ordonnancement utilise le mécanisme de backoff exponentiel pour minimiser le polling des esclaves inactifs de telle façon que la fenêtre de polling d'un esclave qui envoie un paquet NULL est doublé et elle est remise à 1 dans le cas contraire.
Ainsi, cette méthode de polling introduit une certaine QoS. Cependant, elle n'est pas déterministe dans le sens qu'un esclave "best effort" peut attendre infiniment sans transmettre ses paquets.

HOL Priority and HOL K-Fairness Scheduling Kalia et al. ont introduit deux méthodes de polling étiquetées "Master-Slave Queue-State-Dependent Packet Scheduling policies" [23, 24]. Ces méthodes supposent la connaissance du paquet en tête des files (head-of-line, HOL) du maître et de l'esclave. Les paires maître-esclave sont classées en différents classes selon le nombre de slots perdu durant le service de l'esclave appartenant à une connexion maître-esclave donnée. La première classe dont la priorité est la plus haute regroupe les couples de paquets $M \rightarrow S$ et $S \rightarrow M$ où aucun slot n'est perdu. La deuxième classe regroupe les couples de paquets avec un taux de gaspillage inférieur à 25% (par exemple, une transmission d'un paquet de 3 slots suivis d'un paquet Null). La troisième classe dont la priorité est la plus basse, regroupe les couples de paquets entraînant plus de 50% de gaspillage (par exemple pour la transmission d'un paquet de 1 slot, suivi d'un message Null).
En considérant ces classes, Kalia et al. ont définit deux méthodes de polling des esclaves :

1. **HOL Priority Policy (HOL-PP) :** les paires maître-esclave sont servis selon Weighted Round Robin (WRR) où chaque paire a une priorité qui dépend de sa classe. Le principal inconvénient de cette méthode de polling est que la transmission d'un couple de paquets $M \rightarrow S$ et$S \rightarrow M$ de faible prio-

rité n'est effectué qu'après la transmission de tous les paquets appartenant aux classes supérieures. Ce qui peut entraîner des délais importants pour les paquets de faible priorité.

2. **HOL K-Fairness Policy (HOL-KFP) :**les paires maître-esclave sont servis selon Weighted Round Robin (WRR). Cependant, le nombre de polls successifs de la classe de plus haute priorité possède une borne supérieure. Cette borne permet résoudre le problème de HOL-PP qui peut engendrer des délais importants pour les paquets de plus faible priorité. Pour KFP, le maître maintient un compteur pour chaque couple de messages $M \rightarrow S$ et $S \rightarrow M$. Ce compteur comptabilise l'excès ou le déficit de service pour chaque couple et est borné pour assurer une certaine équité. Quand le compteur parcoure K valeurs différentes, le couple ayant la plus basse valeur du compteur est pollé. Par conséquent, le paramètre K représente l'iniquité maximale autorisée (en termes de slots) entre deux couples $M \rightarrow S$ et $S \rightarrow M$ de classe différente.

Adaptive Capacity Ratio with Intelligent Frame Scheduling (ACRIFS) : cet d'ordonnancement prédit l'état des files d'attentes des esclaves afin de leur assigner les capacités nécessaires. Le maître maintient une file d'attente par esclave. Les capacités des files sont calculées périodiquement, en se basant sur des équations faisant intervenir le trafic et le délai de propagation du lien. L'ordonnanceur sert la file d'attente de longueur maximale en premier.

Class-Based Earliest Deadline First (CB-EDF) : Cette technique [50] utilise l'algorithme 1-RR amélioré comme méthode d'ordonnancement local tout en étendant la prise en compte de classes et des échéances des paquets au niveau intra-piconet. En considérant les états des files $M \rightarrow S$ et $S \rightarrow M$, le maître choisit l'esclave ayant la classe la plus prioritaire. Si plusieurs esclaves se portent candidats, le maître choisit celui ayant le paquet avec l'échéance absolue la plus proche. Pour avoir une connaissance sur les états des files $S \rightarrow M$, le maître utilises des mécanismes de signalisation et de prédiction en utilisant les bits "FLOW" et "SEQN" du paquet Bluetooth. Cependant, le bit "FLOW" est conçu pour le contrôle de flux. De même, le bit "SEQN" est conçu pour le séquencement des paquets Bluetooth et ne peut pas, donc, être utilisé pour la signalisation. De plus, CB-EDF est basée sur le mécanisme de polling 1-RR, qui n'assure pas une équité entre les esclave dans le cas de trafics non uniforme

1.7.5.5 Comparaison des ordonnancements intra-piconet

Dans cette section, nous comparons, dans les tableaux 1.1, 1.2, 1.3 et 1.4, les solutions d'ordonnancement intra-piconet selon les critères suivants :

1. **Ordre du polling :** le maître peut poller les esclaves d'une manière **statique** ou **dynamique**. Pour le cas statique, tous les esclaves du piconet sont pollés, même s'ils n'ont rien à transmettre. Pour le cas dynamique, les esclaves du

piconet sont pollés en fonction de l'état de leurs files d'attente de telle façon que ceux ayant des files d'attente vides sont moins pollés pour éviter le gaspillage de la bande passante.

2. **Priorité :** une priorité statique ou dynamique peut être assigné aux esclaves. Cette priorité permet de déterminer l'ordre de polling. Si elle est statique, les esclaves sont pollés dans un ordre fixe tout au long de la durée de vie du réseau. Si elle est dynamique, l'ordre de polling des esclaves peut varier d'un cycle à un autre.
3. **Connaissance des files** $S \rightarrow M$ **:** le maître ne peut connaître que les états des files $M \rightarrow S$. Cependant, des mécanismes de signalisation et de prédiction peuvent être utilisés pour connaître les états des files $S \rightarrow M$.
4. **Mécanismes de signalisation :** des mécanismes de signalisation peuvent être utilisés pour permettre au maître de connaître le statut des files $S \rightarrow M$.
5. **Mécanismes de prédiction :** le maître peut utiliser des mécanismes de prédiction pour estimer la présence de paquets dans les files $S \rightarrow M$.
6. **Intervalle de polling :** l'intervalle de polling peut être **statique** (un esclave est pollé périodiquement) ou **dynamique** (il diminue avec l'augmentation de l'activité de l'esclave ou de la longueur des files $M \rightarrow S$ ou $S \rightarrow M$). Il peut aussi être **déterministe** ou **non déterministe**.
7. **Nombre de paquet transmis :** il représente le nombre de paquets qui peuvent être transmis avant de poller l'esclave suivant. Il peut être égal à 1 ou à un seuil particulier ou exhaustif (ne passer à l'esclave suivant qu'après avoir vidé les files $M \rightarrow S$ et $S \rightarrow M$ de l'esclave courant).

TAB. 1.1 – Mécanismes de polling basés sur Round Robin

	PRR	ERR	EPM	LRR	LWRR
Ordre du polling	statique	statique	dynamique	statique	dynamique
priorité	N.A	N.A	longueur de la file $M \rightarrow S$	N.A	longueur de la file $M \rightarrow S$
Connaissance des files $S \rightarrow M$	non	non	non	non	non
Signalisation	non	non	non	non	non
Prédiction	non	non	non	non	non
Intervalle de polling	statique	dynamique selon l'activité de l'esclave	dynamique selon l'activité de l'esclave	statique	dynamique selon l'activité de l'esclave
Nombre de paquets transmis	1	exhaustif	exhaustif	seuil	seuil

TAB. 1.2 – Mécanismes de polling améliorant Round Robin

	PLsWRR	DRR	RR-FCFS	LA	LARR
Ordre du polling	dynamique	dynamique	dynamique	dynamique	dynamique
priorité	dynamique (échange POLL-NULL)	Taille des données de la file $M \rightarrow S$	FCFS	taille des paquets en tête des files $M \rightarrow S$ et $S \rightarrow M$ combinées	N.A
Connaissance des files $S \rightarrow M$	non	non	non	oui	oui
Signalisation	non	non	non	oui	oui
Prédiction	non	non	non	non	non
Intervalle de polling	dynamique déterministe	dynamique déterministe	dynamique non déterministe	dynamique non déterministe	dynamique déterministe
Nombre de paquets transmis	seuil	1	limité selon FCFS	limité par la priorité	1

TAB. 1.3 – Mécanismes de polling adaptatifs (1/2)

	FEP	**PFP**	**EDC**	**AFP SAFP**
Ordre du polling	statique	dynamique	dynamique	statique
priorité	N.A	bande passante allouée	N.A	longueur de la file $S \rightarrow M$
Connaissance des files $S \rightarrow M$	non	oui	non	oui
Signalisation	non	non spécifié	non	bit flow
Prédiction	non	non	non	non
Intervalle de polling	statique	statique	dynamique	dynamique
Nombre de paquets transmis	1	fraction	1	1(AFP) seuil (SAFP)

TAB. 1.4 – Mécanismes de polling adaptatifs (2/2)

	HOL-PPHOL-KFP	**PS-EXP**	**ACRIFS**	**CB-EDF**
Ordre du polling	dynamique	dynamique	dynamique	dynamique
priorité	classe de la file $M \rightarrow S$	délai tolérable des paquets	longueur des files d'attente $S \rightarrow M$	classe et échéance absolue des paquets
Connaissance des files $S \rightarrow M$	oui	non	oui	oui
Signalisation	non specifié	non	non	oui
Prédiction	non	non	oui	oui
Intervalle de polling	N.A	dynamique non déterministe	dynamique non déterministe	dynamique non déterministe
Nombre de paquets transmis	1	dynamique selon le délai des paquets	dynamique selon la taille des files	1

1.7.6 État de l'art de l'ordonnancement inter-piconet

Dans un piconet, le maître suppose que l'esclave est toujours disponible. Cependant, un esclave peut appartenir à deux piconets (passerelle). Ainsi, des conflits peuvent avoir lieu si le maître polle une passerelle alors que ce dernier est engagé dans un autre piconet. Pour éviter ces conflits, l'ordonnancement inter-piconet est utilisé. Il permet de coordonner les activités des passerelles qui sont présentes dans plusieurs piconets. En conséquence, l'algorithme d'ordonnancement inter-piconet doit s'assurer de la disponibilité des esclaves quand le maître veut communiquer avec eux. Ainsi, il doit déterminer sur quel piconet une passerelle doit être présente à un instant donnée. Un algorithme d'ordonnancement inter-piconet doit donc répondre aux questions suivantes :

1. Comment savoir la présence d'une passerelle sur un piconet ?
2. Comment un maître connaît les périodes de présence d'une passerelle dans son piconet ?
3. Comment l'ordonnancement intra-piconet va tenir compte de l'algorithme d'ordonnancement inter-piconet ?
4. Comment réagir aux changement de topologie ?

Pour répondre aux deux première question, nous distinguons deux approches : approche centralisée et approche distribuée. L'approche centralisée détermine, à la formation du scatternet, les intervalles de présence de chaque passerelle sur un

piconet donnée. Un tel ordonnancement nécessite une synchronisation globale des passerelles sur le scatternet et ne ne permett pas une adaptation aux changement de topologie et de trafic. L'approche distribuée se base sur un algorithme distribuée sur les maîtres et les passerelles des piconets adjacents. Chaque maître se coordonne uniquement avec les passerelles appartenant à son piconet. L'etat de l'art montre que la plupart des solutions d'ordonnancement inter-piconet proposées dans la littérature se basent sur l'approche distribuée [56–59]. Ces solutions s'intéressent à la détermination des instants de présence d'une passerelle sur chacun des piconets auxquels elle est attachée et peuvent être classées, selon la technique utilisée, en deux catégorie : solutions utilisant la technique de Rendez-Vous (RV) et solutions utilisant la technique de Points de Présence (PP).

1.7.6.1 Ordonnancement inter-piconet basée sur la technique de Rendez-Vous (RV)

Un Rendez-Vous est un slot de temps sur lequel une passerelle et un maître auquel elle est attachée se sont mis d'accord. Pour ce slot, le maître "polle" la passerelle qui l'écoute. L'utilisation de la technique de Rendez-Vous nécessite un mécanisme de signalisation spécifique, utilisé pour échanger les informations de contrôle entre les deux stations. Dans la littérature, plusieurs solutions utilisant la technique de Rendez-Vous ont été proposées pour l'ordonnancement inter-piconet [56–60].

Johansson et al. [56] ont proposé une algorithme d'ordonnancement où les Rendez-Vous sont déterminés d'une manière périodique. Cet algorithme utilise des trames spéciales, commun à tous les noeuds, et utilisées pour spécifier la période de temps entre deux points Rendez-Vous pour n'importe quel paire (maître, passerelle). Cet algorithme a été amélioré par Kazantzidis et al. [57] en choisissant les points Rendez-Vous optimaux tout en tenant compte de la mobilité. Dans [58], les points de Rendez-Vous sont déterminés sans aucun message de signalisation. l'algorithme assigne les points de Rendez-Vous selon une séquence pseudo aléatoire spécifique à chaque paire (maître passerelle). Ainsi, les collisions (conflits) entre les points de Rendez-Vous sont réduites. Cependant, cela n'est vrai que pour les densités faibles. Tan et al. ont proposé un algorithme permettant de résoudre le problème de conflits [59] mais en introduisant un overhead. Cet algorithme est basé sur les points de rendez-Vous pour lesquels, les paires (maître-passerelle) se rencontrent pour échanger des données. A la fin de chaque rencontre, les noeuds négocient la date et la durée du prochain rendez-vous.

Contrairement aux algorithmes que nous venons de décire, Zhang et al. [60] ont proposé un algorithme d'ordonnacement inter-piconet qui tient compte du trafic et des poll perdus. Cet algorithme se base sur une table de commutation créée à la création du scatternet. Cette table permet à un bridge de commuter entre les maîtres des piconets auquels il est attaché et elle est mise à jour dynamiquement, selon le trafic et les poll perdus.

1.7.6.2 Ordonnancement inter-piconet basée sur la technique des Points de Présence (PP)

Un Point de Présence est un instant pour lequel une passerelle est censée être présente sur un piconet. La principale différence avec les Rendez-Vous est que le Point de Présence n'implique pas un accord explicite entre le maître et la passerelle pour être présents à la date négociée. L'échange d'informations de contrôle n'est, donc, pas obligatoire pour définir un Point de Présence.

La technique des Points de Présence est utilisée, soit par la passerelle soit par le maître. Dans le premier cas, la passerelle peut imposer ses instants de présence aux maîtres des piconets qui ne la polleront que lorsqu'elle est présente sur le piconet correspondant. Dans le second cas, les instants de polling de la passerelle sur chacun des piconets sont imposés par les maîtres respectifs.

Baatz et al. [61, 62] ont proposé un algorithme d'ordonnancement utilisant le concept de Points de Présence. Ces points dont définis où la communication entre un maître et une passerelle peut commencer. La longueur de cette communication n'est pas fixé à l'avance mais dépend de l'utilisation courante des liens et de la taille des données à échanger. De plus, pour augmenter la densité des Point de présence, les auteurs de [61, 62] ont intégré les Points de Présence avec les slots "sniff". Ces derniers sont vues comme des Points de Présence.

1.7.6.3 Signalisation

Pour signaler les intervalles de sa présence aux maître des piconets auxquels elle est rattachée, une passerelle doit se coordonner avec les maîtres correspondants en utilisant, généralement, des messages de signalisation. Les spécifications Bluetooth suggèrent pour l'ordonnancement inter-piconet d'utiliser la signalisation des mécanismes de conservation d'énergie. Ils permettent à la passerelle d'exprimer au maître ses intervalles de présence sur un piconet. Cependant, la détermination de la présence d'une passerelle sur un piconet peut nécessiter l'échange d'informations de contrôle entre le maître et la passerelle concernée. Un accord explicite entre le maître et une passerelle indique que les deux entités se sont mis d'accord sur un instant (ou un slot) où la passerelle sera présente sur le piconet et le maître viendra la poller. Cet accord nécessite l'échange d'informations de contrôle entre les deux stations pour les synchroniser. Un accord implicite entre le maître et une passerelle ne nécessite pas l'échange d'informations de contrôle. Le maître peut déduire les intervalles d'absence de la passerelle en fonction de paramètres négociées préalablement entre eux.

1.8 Conclusion

Dans ce chapitre, nous avons présenté les réseaux Bluetooth et la problématique d'ordonnancement dans de tel réseaux. Nous avons montré que les mécanismes de communication de Bluetooth, tel que présenté dans sa spécification, ne proposent aucune différenciation de service pour permettre d'utiliser Bluetooth pour des des