

Title	Bluetooth ネットワークの有線拡張によるホームネットワークの構築
Author(s)	井波, 政朗
Citation	
Issue Date	2004-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/1782
Rights	
Description	Supervisor:丹 康雄, 情報科学研究科, 修士

修 士 論 文

Bluetooth ネットワークの有線拡張による
ホームネットワークの構築

北陸先端科学技術大学院大学
情報科学研究科情報システム学専攻

井波 政朗

2004年3月

修 士 論 文

Bluetooth ネットワークの有線拡張による ホームネットワークの構築

指導教官 丹康雄 助教授

審査委員主査 丹康雄 助教授
審査委員 篠田陽一 教授
審査委員 日比野靖 教授

北陸先端科学技術大学院大学
情報科学研究科情報システム学専攻

210005 井波 政朗

提出年月: 2004 年 2 月

概要

近年, 家庭内の機器をネットワークを介して相互に接続するホームネットワークが一般家庭に導入されようとしている. 短距離無線通信技術である Bluetooth[1] は, ホームネットワーク構築のためのインフラ技術として注目される技術の1つである. 本論文では, Bluetoothを用いたホームネットワーク構築の際に問題となる, 電子レンジや無線 LAN などからの干渉, Bluetooth 接続範囲外の機器間の接続に対して, Bluetooth ネットワークを有線伝送路を用いて拡張するシステムを提案する. 提案システムにより, 家庭内における Bluetooth 機器間の安定した通信の提供, および接続範囲の柔軟な拡張を実現する. Bluetooth 規格における HCI データフレームおよび HCI イベントを転送することで特別なプロファイルを持たない既存の Bluetooth 機器間を透過可能な有線接続システムを提案し, 有線接続システムの設計, 実装および評価を行う.

目次

第1章	はじめに	1
1.1	研究の背景	1
1.2	研究の目的	1
1.3	本論文の構成	2
第2章	Bluetooth	4
2.1	Bluetooth 技術仕様	4
2.2	Bluetooth 通信技術	4
2.2.1	マスタ - とスレーブ	4
2.2.2	周波数ホッピング	4
2.2.3	時分割スロット多重	5
2.2.4	ピコネット内同期	6
2.3	Bluetooth デバイスアドレス	6
2.4	通信リンク	6
2.4.1	ACL リンク	6
2.4.2	SCO リンク	7
2.5	パケットタイプ	7
2.5.1	ACL パケット	7
2.5.2	SCO パケット	8
2.6	Bluetooth プロトコルスタック	9
2.6.1	Bluetooth プロトコルスタック	9
2.6.2	物理層 (RF)	9
2.6.3	ベースバンド層 (Baseband)	9
2.6.4	リンク管理層 (Link Manager)	10
2.6.5	論理リンク管理層 (L2CAP:Logical Link Control and Adaptation Protocol)	10
2.6.6	プロファイル層 (Profile)	11
2.6.7	アプリケーション層 (Application)	11
2.7	HCI (Host Control Interface)	11
2.7.1	HCI コマンドパケット	12
2.7.2	HCI イベントパケット	12

2.7.3	HCI データパケット	13
第 3 章	接続手法の検討	14
3.1	物理層での相互接続	14
3.2	データリンク層での相互接続	15
3.3	プロファイルによる相互接続	16
3.4	提案する接続手法	16
第 4 章	提案システム	18
4.1	有線接続システム	18
4.2	Bluetooth インターフェース	18
4.3	Manager	19
4.3.1	イベントの転送	19
4.3.2	データパケットの転送	21
4.4	提案システムのプロトコルスタック	22
第 5 章	提案システムの課題と解決手法	23
5.1	ピコネット内同期	23
5.2	リモートの Bluetooth 機器情報の取得	27
5.3	Bluetooth デバイスアドレス	27
5.4	Bluetooth インターフェースの MTU(Max Transfer Unit)	28
5.5	ローカル側リンクとリモート側リンクで使用するパケットタイプの一致	29
5.6	有線伝送路でのデータの保証	30
5.7	接続認証	30
5.8	接続セットアップ時のセキュリティ・モードの一致	30
第 6 章	メッセージ・シーケンス・チャート	32
6.1	Bluetooth デバイス情報の取得	32
6.1.1	情報取得	32
6.1.2	Bluetooth デバイス情報の反映	34
6.2	ACL 接続要求フェーズ	35
6.3	認証および暗号化	39
6.3.1	ACL 切断	45
6.4	ACL 接続確立後のアクティビティ	46
6.4.1	認証の要求	46
6.4.2	接続の暗号化の設定	47
6.4.3	接続リンク・キーの変更	48
6.4.4	マスター・リンク・キー	49
6.4.5	QoS のセットアップ	50

6.4.6	役割の切り替え	51
6.5	SCO 接続の確立と切断	52
6.5.1	SCO 接続セットアップ	52
6.5.2	SCO 切断	53
6.6	特別なモード	54
6.6.1	Sniff モード	54
6.6.2	Hold モード	57
6.6.3	Park モード	59
第 7 章	提案システムの実装	62
7.1	実装	62
7.2	既存の Bluetooth 機器に対する接続検証	64
7.3	実装した提案システムの評価	65
7.3.1	コネクション確立までの遅延	65
7.3.2	データパケットの伝送遅延	67
7.3.3	スループット	69
7.3.4	実装した提案システムに関する考察	75
第 8 章	提案システムの評価	77
8.1	距離に対する提案システムの有効性	77
8.2	干渉に対する提案システムの有効性	83
8.3	通信リンクに関するパラメータによる評価	88
第 9 章	考察と今後の課題	90
9.1	提案システムの有効性	90
9.2	提案システムの実装に関して	90
9.3	提案システムを用いる有線伝送路に関して	90
9.3.1	有線伝送路間のプロトコル	90
9.3.2	有線伝送路の通信メディア	91
9.4	Bluetooth 規格に関して	91
9.4.1	ピコネット内同期	92
9.4.2	リモート側の Bluetooth 機器情報の取得	92
9.4.3	Bluetooth デバイスアドレス	92
9.4.4	パケットタイプの一致	93
9.4.5	理想的な有線接続システム	93
9.5	他の手法との比較	94
第 10 章	まとめ	95

付録 A	提案システムの関数紹介	98
A.1	全体構成	98
A.2	Bluetooth Interface Handler	98
A.2.1	open_hci_socket	98
A.2.2	hci_open	99
A.2.3	hci_close	100
A.3	Wired Interface Handler	100
A.3.1	wired_open	100
A.3.2	wired_close	100
A.4	Manager	101
A.4.1	wb2_init	101
A.4.2	wb2_collect_remote_bd_info	101
A.4.3	int wb2_send_local_bd_info	102
A.4.4	wb2_recv_remote_bd_info	102
A.4.5	wb2_set_bd_info	103
A.4.6	wb2_event_loop	103
A.4.7	send_hci_cmd_pkt	104
A.4.8	recv_hci_evt_pkt	104
A.4.9	send_hci_acl_pkt	105
A.4.10	recv_hci_acl_pkt	105
A.4.11	send_hci_sco_pkt	106
A.4.12	recv_hci_sco_pkt	106

目次

1.1	有線拡張システムのイメージ	3
2.1	ピコネット	5
2.2	周波数ホッピング	5
2.3	周波数分割多重スロットとタイミング	6
2.4	Bluetooth プロトコルスタック	10
2.5	Bluetooth ソフトウェア・スタックの下位層	11
2.6	HCI コマンドパケット	12
2.7	HCI イベントパケット	12
2.8	HCI ACL データパケット	13
2.9	HCI SCO データパケット	13
3.1	伝送信号の中継による有線接続	14
3.2	マスターの送信/受信タイミング	15
3.3	HCI データパケットの転送による有線接続	16
4.1	提案システム	18
4.2	HCI コマンドによる HCI イベントの転送	19
4.3	ローカル側における接続要求の受信	20
4.4	ローカル側からの接続要求の受信	20
4.5	データパケットの転送	21
4.6	有線接続システムのプロトコルスタック	22
5.1	ピコネット内同期問題	23
5.2	ピコネット内同期問題	25
5.3	ピコネット内同期問題の解決機構	26
5.4	有線接続システムの接続の仕組み	28
5.5	パケットタイプの相違	29
6.1	Bluetooth デバイス情報の取得	33
6.2	Bluetooth デバイス情報の反映	34
6.3	ACL 接続要求フェーズ 1	36
6.4	ACL 接続要求フェーズ 2	37

6.5	ACL 接続要求フェーズ 3	38
6.6	リモート側の Bluetooth 機器がペアリング・認証を要求	41
6.7	リモート側の Bluetooth 機器が暗号化を要求	42
6.8	ローカル側の Bluetooth 機器が認証を要求	43
6.9	ローカル側の Bluetooth 機器が暗号化を要求	44
6.10	ACL 切断	45
6.11	認証の要求	46
6.12	接続の暗号化の設定	47
6.13	接続リンク・キーの変更	48
6.14	マスター・リンク・キー	49
6.15	QoS のセットアップ	50
6.16	役割の切り替え	51
6.17	SCO 接続セットアップ	52
6.18	SCO 切断	53
6.19	Sniff モード	55
6.20	Sniff モード 終了	56
6.21	Hold モード	58
6.22	Park モード	60
6.23	Park モードの終了	61
7.1	実装した提案システム	62
7.2	実装した提案システムの実装環境	63
7.3	実装した提案システムのソフトウェア構成	63
7.4	提案システムと Bluetooth 機器の配置図	65
7.5	提案システムの遅延	68
7.6	提案システムのスループット (DM1 パケット使用時)	70
7.7	提案システムのスループット (DH1 パケット)	71
7.8	提案システムのスループット (DM3 パケット使用時)	72
7.9	提案システムのスループット (DH3 パケット使用時)	73
7.10	提案システムのスループット (DM5 パケット使用時)	74
7.11	提案システムのスループット (DH5 パケット使用時)	75
8.1	距離に対する提案システムの評価環境	78
8.2	接続距離に対するスループットの比較 (DM1 パケット使用時)	79
8.3	接続距離に対するスループットの比較 (DH1 パケット使用時)	80
8.4	接続距離に対するスループットの比較 (DM3 パケット使用時)	80
8.5	接続距離に対するスループットの比較 (DH3 パケット使用時)	81
8.6	接続距離に対するスループットの比較 (DM5 パケット使用時)	81
8.7	接続距離に対するスループットの比較 (DH5 パケット使用時)	82

8.8	干渉に対する提案システムの評価環境	84
8.9	干渉を受ける環境におけるスループットの比較 (DM1 パケット使用時)	85
8.10	干渉を受ける環境におけるスループットの比較 (DH1 パケット使用時)	85
8.11	干渉を受ける環境におけるスループットの比較 (DM3 パケット使用時)	86
8.12	干渉を受ける環境におけるスループットの比較 (DH3 パケット使用時)	86
8.13	干渉を受ける環境におけるスループットの比較 (DM5 パケット使用時)	87
8.14	干渉を受ける環境におけるスループットの比較 (DH5 パケット使用時)	87
8.15	通信リンクに関するパラメータの測定環境	88
A.1	実装した提案システムの全体図	98

表 目 次

4.1	転送を必要とする HCI イベント	21
7.1	直接接続した場合との評価環境	65
7.2	L2CAP コネクション確立までの平均時間	66
7.3	L2CAP コネクション確立までの測定された最大時間	66
7.4	ACL パケットの平均データ長と 1 秒間に処理されるパケット数	70
8.1	距離に対する提案システムの評価	77
8.2	干渉に対する提案システムの評価	83
8.3	通信リンクに関するパラメータの測定結果	89
9.1	有線接続手法の比較	94

第1章 はじめに

1.1 研究の背景

近年, ネットワーク技術の進歩, 情報家電, 情報端末自身の技術的な進歩によって, 家庭の機器が情報化され, それらがネットワークを介して接続されるホームネットワークの構築が実現可能になりつつある. これにともなって, エアコンなどの白物家電の制御を目的とした Echonet[2], PC や比較的高度な情報家電の制御を目的とした UPnP[3] 等, ホームネットワークを構築するための様々なミドルウェアが検討され, また, これらのインフラとなるネットワーク技術に関しても Bluetooth, 無線 LAN[4], UWB(Ultra Wideband)[5] など様々な技術が提案されている. 一般に, 家庭内におけるネットワークには, 新たな配線が不要であること, ネットワークに参加, 離脱するための操作が容易であること, 低消費電力, 高セキュリティなどオフィスにおけるネットワークとは異なった技術が要求される. このような要求に対し Bluetooth は, アド・ホックにネットワークを構築可能な点や, 低消費電力, セキュリティなどの要件が規格に含まれている点など, ホームネットワーク構築のためのインフラ技術として期待されるネットワーク規格の1つである.

1.2 研究の目的

Bluetooth を用いた家庭内ネットワークの構築には, 電子レンジや無線 LAN などの Bluetooth と同一の周波数帯を利用する機器から干渉を受ける場所においても安定した接続を提供できること, 離れた場所, 壁などの障害物を挟んで存在する Bluetooth 接続範囲外となる機器間の相互接続を提供できることが課題となる. このような課題に対して, 本研究では, Bluetooth ネットワークを有線拡張する接続システムを提案する(図 1.1). Bluetooth ネットワークを拡張することで, 特別なプロファイルを持たない既存の Bluetooth 機器を透過に接続可能にし, ノイズや干渉を受ける場所における安定した機器間の接続, Bluetooth 接続範囲の柔軟な拡張を可能にする. 提案する有線接続システムの実装を行い, 提案システムの有効性をスループット, リンクの安定性などの点から検証を行う.

Bluetooth ネットワークを有線拡張することで, 特別なプロファイルをもたない既存の Bluetooth 機器間に対しても透過に有線接続可能な接続システムを実現し, ノイズや干渉を受ける場所における場所における Bluetooth 機器間の安定した接続, Bluetooth 接続範囲の柔軟な拡張を可能にする.

1.3 本論文の構成

本論文は以下の構成になっている。

- 第1章 … 研究の背景と目的, 本論文の全体の流れを説明する.
- 第2章 … 本研究の研究対象である Bluetooth 規格に関して通信技術などをまとめる.
- 第3章 … Bluetooth ネットワークを有線拡張するための手法に関して物理層, データリンク層, プロファイルによる手法をそれぞれについて検討し, 本研究で提案する有線接続手法について述べる.
- 第4章 … 本論文で提案する Bluetooth ネットワークを有線接続するシステムについて説明する.
- 第5章 … 提案システムによる既存の Bluetooth 機器間の接続の際の課題と, 課題を解決するために提案システムが用いる解決機構について説明する.
- 第6章 … 提案システムを用いて Bluetooth 機器間を接続した際におけるメッセージ・シーケンス・チャートを示す.
- 第7章 … 提案システムの実装を行い, 実装した提案システムを用いて提案システムを用いることによる影響を評価を行う.
- 第8章 … 提案システムの有効性に関する評価を行う.
- 第9章 … 提案システムについて考察を行い, 今後課題について述べる.
- 第10章 … 本研究について総括する.

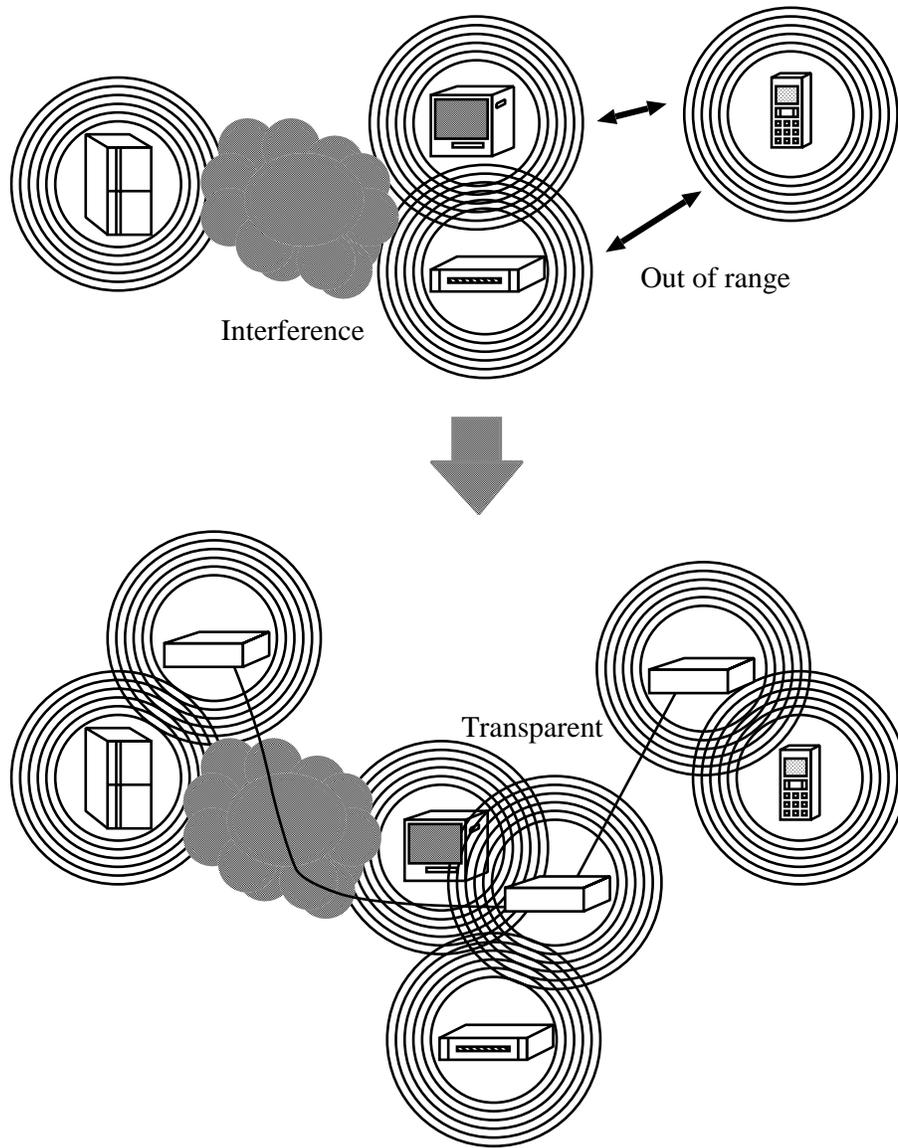


図 1.1: 有線拡張システムのイメージ

第2章 Bluetooth

本章では, Bluetooth 技術仕様 [1] に関して概要をまとめる. Bluetooth 技術仕様において本稿で提案する有線接続システムに関わる Bluetooth の通信技術, Bluetooth デバイスアドレス, 通信リンク, パケットタイプ, プロトコルスタック, および HCI(Host Control Interface) に関してそれぞれ概説する.

2.1 Bluetooth 技術仕様

Bluetooth は, ISM バンド (2.4GHz 帯) を利用し, 短距離間の音声およびデータ通信を実現するオープンな技術仕様である [6]. 低コスト, 低消費電力, 小サイズのデバイスを目的に, Bluetooth SGI(Special Interest Group)[1] によって規格, 査定, 普及促進, 技術管理などが進められている.

2.2 Bluetooth 通信技術

2.2.1 マスタ - とスレーブ

Bluetooth では, 1つのマスタとなる端末と1つ以上のスレーブとなる端末がワイヤレスネットワークを構築し通信を行う. マスタは, コンピュータネットワークのクライアント/サーバにおけるサーバに相当する端末で, Bluetooth 通信におけるワイヤレスネットワークを制御しながら通信を行う. 一方, スレーブは, クライアントに相当する端末で, マスタの制御を受けながら通信を行う. このマスタとスレーブから構成されるネットワークを Bluetooth では, ピコネット (Piconet) とよび, 同一ピコネット内に属する Bluetooth 通信端末は, 周波数軸上, 時間軸上において同期している状態にある. このとき, 1つのピコネット内で同時に通信できるスレーブの数は最大7つまでである. 図 2.1 に, ピコネットのトポロジ構成例を示す.

2.2.2 周波数ホッピング

Bluetooth では, 通信方式として周波数ホッピング型のスペクトル拡散方式が規定されている. Bluetooth では, 79MHz の周波数幅を使って信号のやりとりがなされ, 1MHz の信

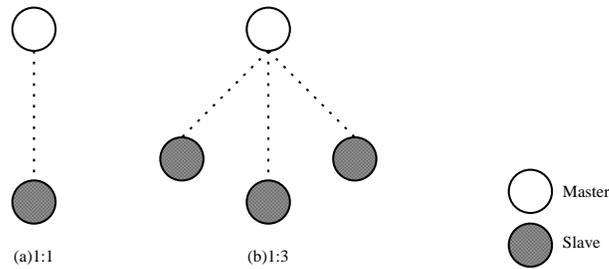


図 2.1: ピコネット

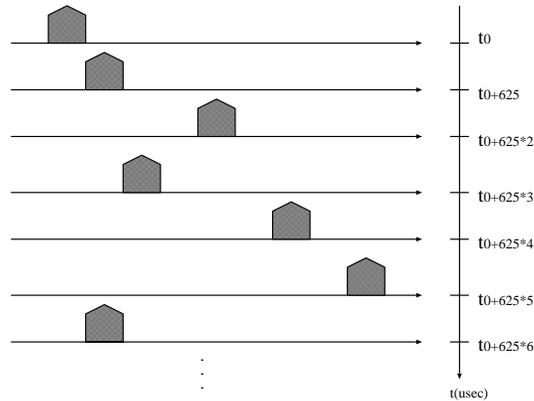


図 2.2: 周波数ホッピング

号周波数帯域を 79MHz 内で $625\mu\text{sec}$ ごとにランダムに変化 (ホッピング) させる (図 2.2). Bluetooth で定義される通信周波数は, 以下のとおりである .

$$\text{Bluetooth 通信周波数} : f(k) = 2402 + k(\text{MHz})$$

$$k = 0, 1, 2, \dots, 78$$

この周波数ホッピングのパターンは, ピコネット内におけるマスターの Bluetooth デバイスアドレスと Bluetooth クロックから一意に算出され, スレーブがホッピングパターンを算出し, 同期することで通信を可能にする .

2.2.3 時分割スロット多重

ピコネットでは, 1つのマスターと最大7つまでのスレーブが同時に通信を行う. 周波数軸上の同期に加えて, スレーブとマスター間の通信路は, それらすべてのスレーブで時分割スロット多重しながら共有される. 分割多重の時間単位は, 時間スロットと呼ばれ $625\mu\text{sec}$ の時間間隔である. 同一ピコネット内に存在するマスターとスレーブの送受信パケットの方向は, 時間スロット番号が偶数の場合, マスターからスレーブに行い, スロット番号が奇

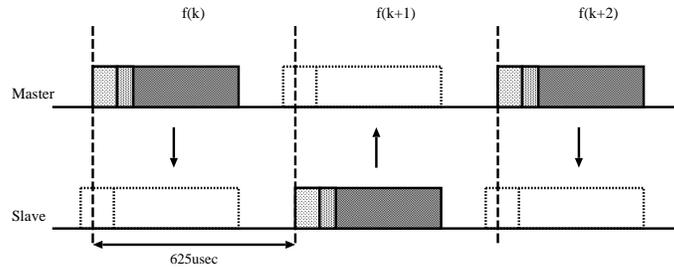


図 2.3: 周波数分割多重スロットとタイミング

数の場合は, スレーブからマスターに行う。

2.2.4 ピコネット内同期

同一ピコネット内に属するすべてのスレーブは, マスターの周波数ホッピングパターンと時間スロットに同期することが通信を行うための条件となる。周波数ホッピングパターンは, ピコネットのマスターの Bluetooth アドレスと Bluetooth クロックから一意的に算出され, すべてのスレーブはマスターと同じ周波数ホッピングパターンに基づいて, 時間スロットごとに通信周波数をホッピングさせる。そして, 時間スロットはマスターの Bluetooth クロックを基準として, 各スレーブが形成する。つまり, ピコネット内で通信接続フェーズにあるすべての端末は, マスターを基準とした同一周波数ホッピングパターンと時間スロットを有している状態にあり, これをピコネット内同期 (Piconet Synchronization) という。

2.3 Bluetooth デバイスアドレス

Bluetooth 機器には, 識別子として 48 ビットの Bluetooth デバイスアドレスが与えられる。このアドレスは, IEEE802 [7] 仕様に準拠したアドレス方式によって定義され, それぞれの端末に対して一意的に与えられる。端末の管理などに用いることを主目的としているが, その一意性から, 周波数ホッピングパターンなどを生成するための重要なパラメータとしても用いられる。

2.4 通信リンク

2.4.1 ACL リンク

ACL リンクは, マスターとピコネットに参加しているすべてのスレーブとの間のポイント・ツー・マルチポイント・リンクである。ACL リンクでは, マスターと, ピコネット

に参加してりるすべてのアクティブなスレーブとの間で接続を確立することができ、非同期および等時同期のサービスがサポートされる。1つのマスターと1つのスレーブの間には、1つのACLリンクだけが存在でき、また、ほとんどのACLパケットでは、データの完全性を保証するための再送が行われる。

2.4.2 SCOリンク

SCOリンクは、マスターと特定の1つのスレーブ間との間の対称型ポイント・ツー・ポイント・リンクである。SCOリンクは、スロットを予約するため、マスターとスレーブ間の回線交換型接続とみなすことができ、音声のような時間に縛られる情報の取り扱いに適している。マスターは、同一スレーブまたは異なるスレーブに対し最大3つのSCOリンクをサポートできる。1つのスレーブは、同一のマスターに対し最大3つのSCOリンクをサポートし、異なるマスターに対し2つのSCOリンクをサポートできる。SCOリンクにおけるマスターとスレーブ間の通信データ速度は64Kbpsであり、またSCOパケットの再送は行われない。

2.5 パケットタイプ

2.5.1 ACLパケット

ACLパケットは非同期リンクで使用され、伝送されるデータはユーザー・データか制御データである。ACLパケットには、7つのパケットが定義されている。このうち6つのパケットにはCRCコードが含まれており、適切な受信を示すアクノレッジを受け取らなければ、ACLパケットの再送が行われる。AUXパケットには、CRCがないため、再送は行われない。

- DM1パケット

DM1パケットは、データ情報だけを運ぶパケットである。DMはデータ・ミディアム(Data-Medium)を意味している。ペイロードは、最大18バイト(1バイトのペイロード・ヘッダを含む)の情報と16ビットのCRCコードからなり、ペイロード・ヘッダの長さは、1バイトである。情報とCRCはレート2/3FECでコード化され、10ビット・セグメントごとに5パリティ・ビットが追加される。DM1パケットは最大で1タイム・スロットを占める。最大データ速度は、順方向(マスターからスレーブ)へ108.8Kbps、逆方向(スレーブからマスター)へ108.8Kbpsである。

- DH1パケット

DH1パケットは、ペイロードの情報がFECでコードされないことを除けば、DM1パケットに似たパケットである。DH1パケットでは、最大28バイトの情報と16ビットのCRCコードが伝送される。DHは、データ・ハイ(Data-High)を意味している。

DH1 パケットは最大で1 タイム・スロットを占める。最大データ速度は、順方向へ 172.8Kbps, 逆方向へ 172.8Kbps である。

- DM3 パケット

DM3 パケットは、DM1 パケットに拡張ペイロードが付加されたものである。ペイロードは、最大 123 バイトの情報 (2 バイトのペイロード・ヘッダを含む) と 16 ビットの CRC コードからなり、ペイロード・ヘッダの長さは、2 バイトである。DM3 パケットは、最大で 3 タイムスロットを占める。最大データ速度は、順方向へ 387.2Kbps, 逆方向へ 54.4Kbps である。

- DH3 パケット

DH3 パケットは、ペイロードの情報が FEC でコードされないことを除けば、DM3 パケットに似たパケットである。DH3 パケットでは、最大 185 バイトの情報と 16 ビットの CRC コードが伝送される。DH3 パケットは最大で 3 タイム・スロットを占める。最大データ速度は、順方向へ 585.6Kbps, 逆方向へ 86.4Kbps である。

- DM5 パケット DM5 パケットは、DM1 パケットに拡張ペイロードが付加されたものである。ペイロードは、最大 226 バイトの情報 (2 バイトのペイロード・ヘッダを含む) と 16 ビットの CRC コードからなり、ペイロード・ヘッダの長さは、2 バイトである。DM5 パケットは、最大で 5 タイムスロットを占める。最大データ速度は、順方向へ 477.8Kbps, 逆方向へ 36.3Kbps である。

- DH5 パケット

DH5 パケットは、ペイロードの情報が FEC でコードされないことを除けば、DM5 パケットに似たパケットである。DH5 パケットでは、最大 185 バイトの情報と 16 ビットの CRC コードが伝送される。DH5 パケットは最大で 3 タイム・スロットを占める。最大データ速度は、順方向へ 723.2Kbps, 逆方向へ 57.6Kbps である。

- AUX1 パケット

AUX パケットは、ペイロードの情報が FEC でコードされないことを除けば、DM1 パケットに似たパケットである。AUX パケットでは、最大 30 バイトの情報を含み、AUX パケットは最大で 1 タイム・スロットを占める。最大データ速度は、順方向へ 185.6Kbps, 逆方向へ 185.6Kbps である。

2.5.2 SCO パケット

SCO パケットは、同期 SCO リンクで使用されるパケットタイプであり、再送されることはない。HV パケットは、主に音声伝送に使用される。いずれのパケットを利用した際にも、最大データ速度は、64Kbps の対称である。

- HV1 パケット
HV1 パケットでは, 10 バイトの情報が伝送される. 1/3 FEC で保護され, ペイロードの長さは 240 ビットの固定である.
- HV2 パケット
HV2 パケットでは, 20 バイトの情報が伝送される. 2/3 FEC で保護され, ペイロードの長さは 240 ビットの固定である.
- HV3 パケット
HV1 パケットでは, 30 バイトの情報が伝送される. 情報は FEC で保護されず, ペイロードの長さは 240 ビットの固定である.
- DV パケット
DV パケットは, データと音声統合されたパケットである. 80 ビットの音声フィールドと最大 150 ビットのデータ・フィールドに分割される. 音声フィールドは, FEC で保護されず, データフィールドは最大 10 バイトの情報 (1 バイトのペイロードヘッダを含む) と 16 ビットの CRC からなる.

2.6 Bluetooth プロトコルスタック

2.6.1 Bluetooth プロトコルスタック

Bluetooth システム全体のプロトコルは, Bluetooth コア, 適合プロトコル, アプリケーションの 3 つのブロックから構成される. Bluetooth のプロトコルスタックを図 2.4 に示す. Bluetooth プロトコルは次の 6 つのプロトコルから構成され, RF 層から L2CAP 層までを Bluetooth コアプロトコルと呼ぶ.

2.6.2 物理層 (RF)

Bluetooth の物理層には, 2.4GHz 周波数帯域における周波数ホッピング型のスペクトラム拡散方式が採用されている. 出力電力は標準 1mW で, およそ 10m の距離で伝播させることができる. また, 最大で 100mW まで送信電力をあげることができ, およそ 100 m の距離まで通信範囲を拡大することができる.

2.6.3 ベースバンド層 (Baseband)

物理層に対して, 実際の送受信データパケットをやり取りするプロトコルとして, Baseband 層が定義されている. 主な役割として, 上位層から受け渡されるデータを送受信するための通信リンクを提供する. また, 物理層に対する周波数ホッピングを管理するための,

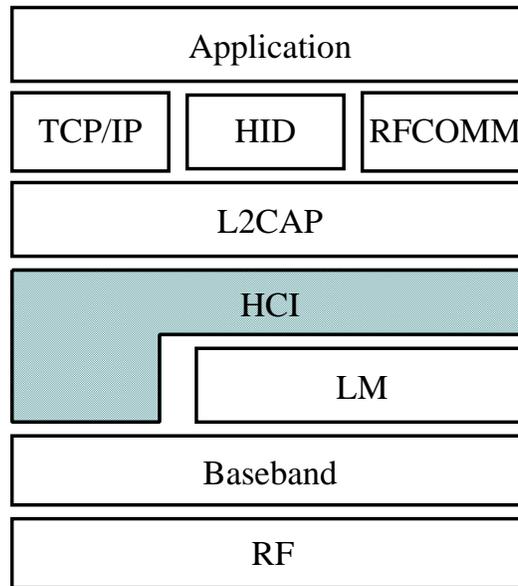


図 2.4: Bluetooth プロトコルスタック

送受信周波数の指定・切り替え制御や時間スロットの管理なども行う。さらに、パケットの再送や誤り訂正, 誤り検出の処理を行う。

2.6.4 リンク管理層 (Link Manager)

リンク管理層は、通信リンク上で送受信パケットをやり取りするプロトコルとして定義されている。Baseband 層で提供される通信リンクの設定や切断など、通信リンクに関わるさまざまな制御機能を LMP(Link Manager Protocol) を用いて提供する。さらに、これらの機能に加えて、通信リンクのセキュリティの管理も行う。

2.6.5 論理リンク管理層 (L2CAP:Logical Link Control and Adaptation Protocol)

論理リンク管理層は、ユーザデータの論理チャンネルの管理を行う。上位アプリケーションからのデータを論理チャンネルとして管理し、データ分割 (フラグメント化) やデータ再構成の処理を行う。また、複数の上位アプリケーションに対するプロトコル多重もこの層において行われる。

2.6.6 プロファイル層 (Profile)

プロファイル層は, Bluetooth コアを既存通信プロトコルを利用するアプリケーションへ適合させることを目的としたプロファイルを定義する. TCP/IP プロトコル, RFCOMM プロトコル, HID (Human Interface Device) などの様々なプロファイルが定義される.

2.6.7 アプリケーション層 (Application)

アプリケーション層は, ユーザのアプリケーションなどユーザに直接サービスを提供する.

2.7 HCI (Host Control Interface)

Host Control Interface (HCI) は, Bluetooth ハードウェア機能にアクセスするための共通インターフェース方式を提供する. 一般的に, Bluetooth プロトコルスタックにおける RF, Baseband, LM のプロトコルは, 1つの Bluetooth モジュールにパッケージ化され, Universal Serial Bus (USB) や RS-232, あるいは UART を経由してホストに接続される. ベンダーが異なっても接続可能な互換性のあるモジュールを開発できるように, Bluetooth 仕様はホストとモジュールを接続する物理インターフェースと独立した, モジュールの低位層にアクセスするための共通インターフェースを定義している. これにより, アプリケーションを含む上位層から単一のインターフェースを用いて, ベースバンドやリンク・マネージャ, その他のハードウェアにアクセスすることが可能となる. 図 2.5 に HCI を含む Bluetooth ソフトウェア下位層の概略を示す.

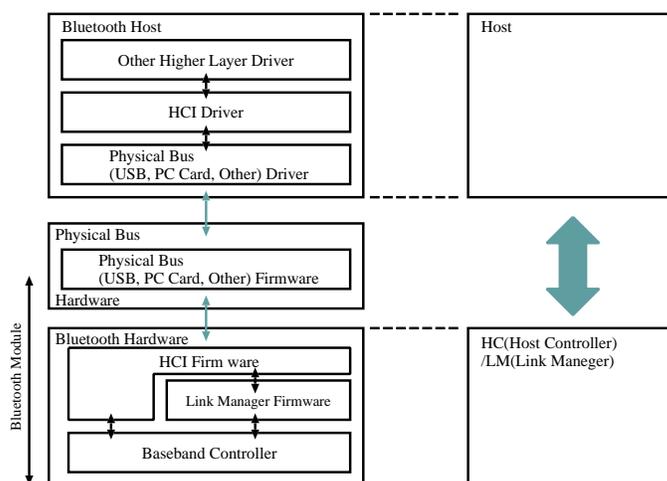


図 2.5: Bluetooth ソフトウェア・スタックの下位層

2.7.1 HCI コマンドパケット

HCI は, Bluetooth ハードウェア機能にアクセスするための一様なコマンド方式を提供する. 上位層のホストが, Bluetooth モジュールを制御する場合は, HCI コマンドを利用し, HCI コマンドパケットは, ホストからホスト・コントローラーにコマンドを送信する際に使用される. HCI コマンドパケットは, 図 2.6 の形式で定義されている. コマンドは, OCF(Opcode Command Field) と OGF(Opcode Group Field) からなる OpCode フィールドに指定され, 各コマンドのパラメータは Parameter フィールドに指定される. コマンドの戻り値は, HCI イベントによって得られる.

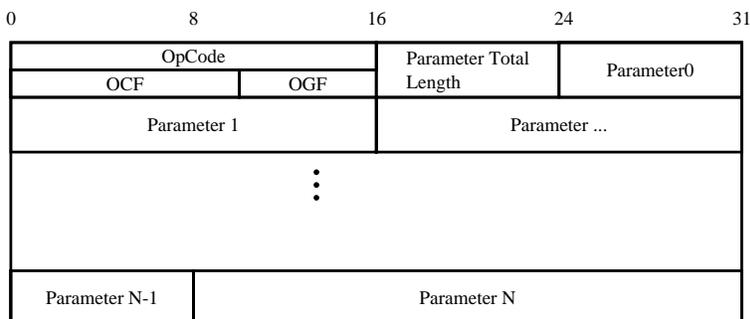


図 2.6: HCI コマンドパケット

2.7.2 HCI イベントパケット

HCI イベントパケットは, ホスト・コントローラーがイベントが発生したことをホストに通知する際に使用される. HCI コマンドパケットは, 図 2.7 の形式で定義されている. イベントの種類は, Event Code フィールドに指定され, 各イベントのパラメータは Parameter フィールドに指定される.

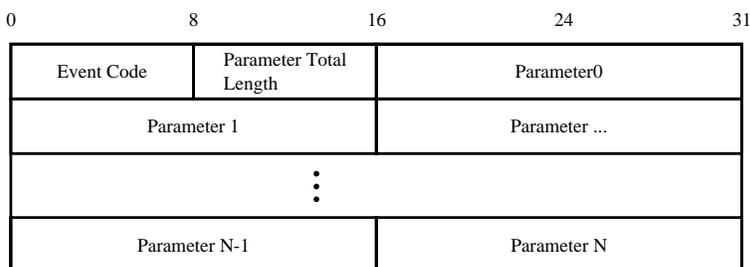


図 2.7: HCI イベントパケット

2.7.3 HCI データパケット

HCI データパケットは、ホストとホスト・コントローラの間でデータを交換する際に使用される。データ・パケットは、ACL と SCO の両方のデータの種類の種類が定義されている。HCI ACL データパケットは図 2.8 の形式、HCI SCO データパケットは図 2.9 の形式で定義される。Connection Handle フィールドには、データパケットの転送に使用される接続ハンドルが指定され、この接続ハンドルにより接続リンクの識別を行う。

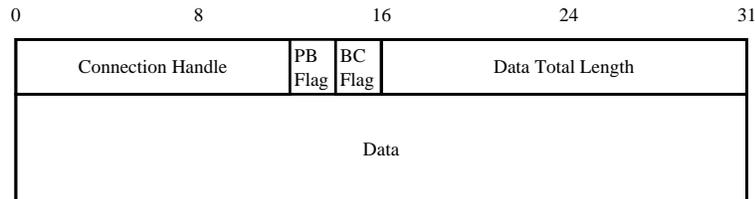


図 2.8: HCI ACL データパケット

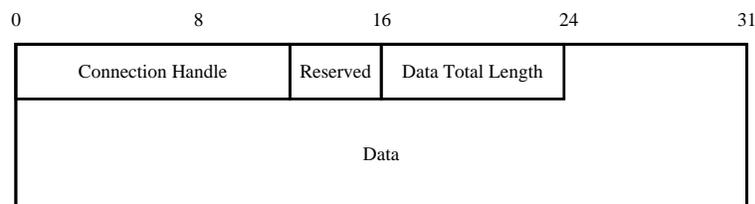


図 2.9: HCI SCO データパケット

第3章 接続手法の検討

本章では、Bluetooth ネットワークの有線拡張方式に関して、物理層、データリンク層、プロファイルの各層による相互接続の実現手法を検討する。検討を行った上で、本稿で提案する有線接続システムが用いる有線拡張方式を提案する。

3.1 物理層での相互接続

Bluetooth ネットワークの有線拡張方式の一つとして、物理層において相互接続を実現する手法が挙げられる。図 3.1 のように RF 層における伝送信号をリピータを用いて有線伝送路で中継することで、Bluetooth 機器間を接続可能な有線接続システムを実現することが可能である。伝送信号を中継することから、この手法を用いることで完全に透過な有線接続システムを実現できると考えられる。

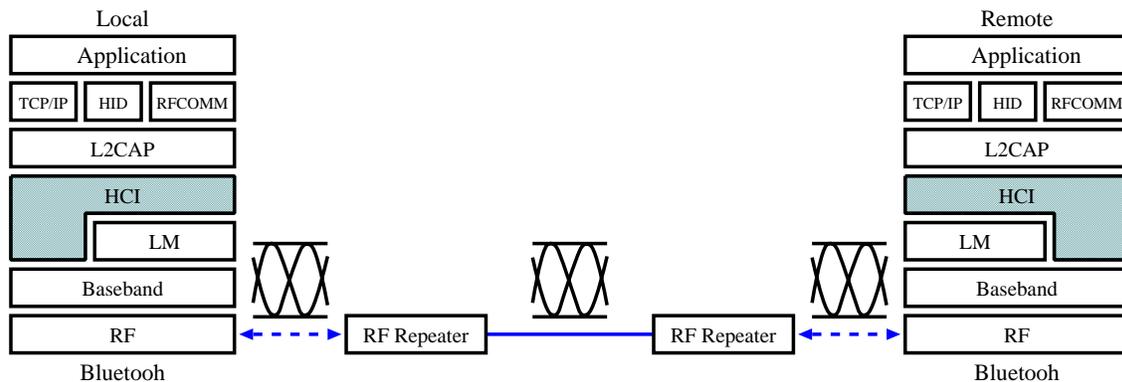


図 3.1: 伝送信号の中継による有線接続

しかし、この手法には、中継する伝送信号の選択に関する問題がある。有線接続システムが中継する伝送信号の選択方法の最も簡単な手法は、Bluetooth の利用する 2.4GHz 帯の信号をそのまま中継する手法であるが、この場合、無線 LAN や電子レンジなどの同一周波数帯を利用する機器からの信号も中継してしまう。これによって、有線接続システムは、干渉をおこす領域を広げることになり、Bluetooth だけでなく無線 LAN など他の通信機器へも影響を与える。この問題に対しては、Bluetooth の伝送信号のみを検出し中継することが考えられるが、この手法の実現には、特殊な回路装置が必要となる。また、検出

などの特殊な処理による遅延は、Baseband 層における通信遅延許容時間に対して大きな問題となる。Bluetooth では、Baseband 層において時分割二重 (TDD) スキームを使用する。これは、送信と受信を同期しながら交互に行う方式である。通常の接続モードでは、マスターの伝送は常に偶数番号タイム・スロットから開始され、スレーブの伝送は常に奇数番号タイム・スロットから開始される。つまり、図 3.2 に示すように、タイム・スロットは $625\mu\text{sec}$ に分割され、マスターがパケットを送信した $625\mu\text{sec}$ 後は必ず受信スロットとなる。タイムスロットの同期を行うため、スレーブはマスターの packets が受信されるたびにパケットを送信するタイミング・オフセットを更新し、送出するパケットが存在する場合は受信した時刻から $625\mu\text{sec}$ 後にパケットを送出する。マスターには、ある程度のタイミングのずれを許容するため、正確な受信タイミングを中心とする不確定ウィンドウが定義されており、RX バーストは最大 $10\mu\text{sec}$ 早いまたは遅い到着まで許容される。つまり、この手法を用いた場合に許容される伝送遅延は最大 $10\mu\text{sec}$ となり、 $10\mu\text{sec}$ 以上の遅延が発生した場合、送信データとして扱われない。このため、有線接続システムの実現には非常に高速な中継能力のある特殊なハードウェアを実装する必要がある。

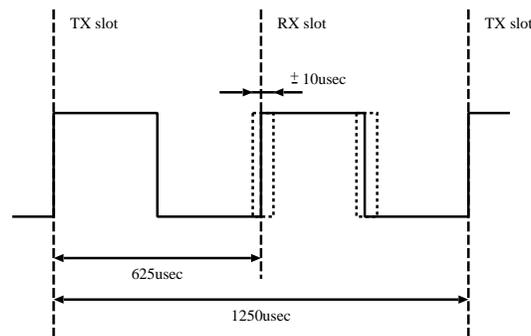


図 3.2: マスターの送信/受信タイミング

3.2 データリンク層での相互接続

Bluetooth ネットワークの有線拡張方式の一つとして、データリンク層において相互接続を実現する手法が挙げられる。図 3.3 のように HCI におけるイベントおよびデータパケットを転送することで、Bluetooth 機器間を接続可能な有線接続システムを実現することが可能である。HCI おけるイベント、データパケットを転送することから、この手法を用いることで L2CAP を含む上位層が透過な有線接続システムを実現できると考えられる。

この手法を用いた場合における問題点は、ローカル側とリモート側でそれぞれ Baseband 層を持つことから、ローカル側、リモート側それぞれで異なるピコネットを構成することになる点である。このため、有線接続システムでは接続される Bluetooth 機器に対して異なるピコネットに接続されていることを同一のピコネットに接続されているように見せ

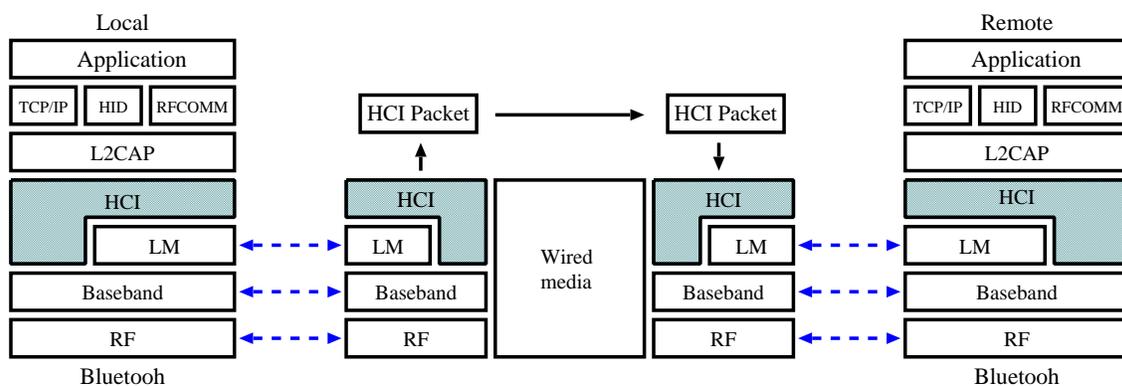


図 3.3: HCI データパケットの転送による有線接続

かけるための処理が必要となる。しかし、ローカル側とリモート側それぞれで異なるピコネットを構成することで、物理層における有線接続の実現の際に問題となる $10\mu\text{sec}$ の通信遅延許容時間に関する問題を回避することが可能となる。また、この手法を用いる場合、有線接続システムは特別なハードウェアを用いなくても、ソフトウェアを用いて実現が可能である。

3.3 プロファイルによる相互接続

Bluetooth ネットワークの有線拡張方式の一つとして、プロファイルによる相互接続を実現する手法が挙げられる。有線メディアを利用する適合プロファイルとして上位層プロトコルスタックを設計することで、Bluetooth 機器同士を有線メディアを利用して相互接続することが可能となる。このような手法を利用した既存の Bluetooth 適合プロファイルの代表的なものとして、LAN アクセス・プロファイルなどが挙げられる。

プロファイルによる有線接続の問題点は、有線接続のための特別なプロファイルをもつ Bluetooth 機器同士のみが接続可能な有線接続システムとなる点である。このためこの手法を用いた場合、設計した有線接続システムを用いて既存の Bluetooth 機器の有線接続を行うことはできない。しかし、プロファイルによる有線接続手法を用いることで、Bluetooth 規格に準拠したシステムが構築可能となる。

3.4 提案する接続手法

本論文では、有線メディアを利用するための特別なプロファイルを持たない既存の Bluetooth 機器を透過に接続可能な点、また、特別なハードウェアを必要とせずに有線接続システムを実現可能点から、データリンク層での相互接続を提供する有線接続システムを提案する。提案する有線接続システムにより、本来、無線のみで構築される Bluetooth ネット

トワークを有線拡張し、機器間の安泰したリンクの提供および Bluetooth 接続範囲の柔軟な拡張が可能になる。

第4章 提案システム

本章では, Bluetooth ネットワークを有線拡張する接続システムを提案し, 接続システムの構成, 接続の仕組み, およびプロトコルスタックについて説明する.

4.1 有線接続システム

本研究で提案する有線接続システムを図 4.1 に示す. 有線接続システムは, ローカル側, リモート側それぞれに Bluetooth に対するインターフェースを持ち, Bluetooth 機器とはこのインターフェースを介して接続する. Bluetooth インターフェースからの HCI イベントおよび HCI データパケットを有線伝送路を用いて転送することで, ローカル側, リモート側の Bluetooth 機器を接続する. L2CAP 層よりも下位に存在する HCI におけるイベントおよびデータパケットを転送することから, 有線接続システムは, 接続対象となる Bluetooth から透過な接続システムとなる.

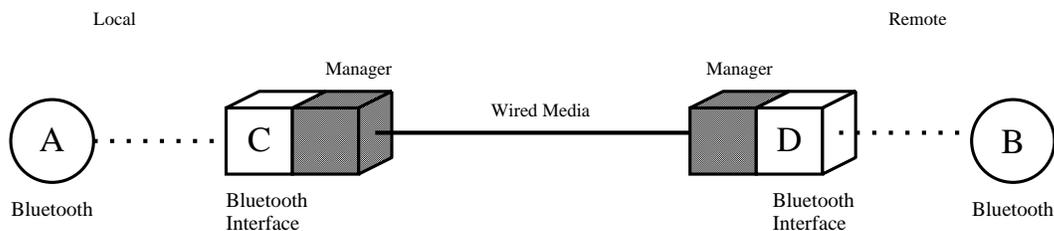


図 4.1: 提案システム

4.2 Bluetooth インターフェース

提案する有線接続システムは, Bluetooth 機器の接続インターフェースとして物理層から, Baseband 層, LM 層, HCI をもつ Bluetooth モジュールを持つ. 有線接続システムでは, Bluetooth モジュールを有線接続システムの持つ Bluetooth インターフェースとし, 複数のインターフェースを持つ場合には, Bluetooth モジュールを複数個接続する. ローカル側, リモート側の Bluetooth インターフェースがそれぞれ Baseband 層を持つことで, ローカル側とリモート側それぞれでピコネットを形成することになる. これによって, 物

理層での接続の際に問題となる $10\mu\text{sec}$ 以内の通信許容遅延時間に関する問題を回避することができる。また、Bluetooth インターフェースはそれぞれ固定の Bluetooth デバイスアドレスを持つ。

4.3 Manager

Manager は、有線伝送路を利用して Bluetooth 機器間の透過接続を実現するための処理を行う。Manager の行う処理機構は次の 2 つに分かれる。

1. HCI イベントおよび HCI データパケットの転送処理を行う機構
2. 提案システムによる透過接続を実現するための課題を処理する機構

本章では、HCI イベントと HCI データパケットの転送処理の仕組みに関して説明し、透過接続を実現するための課題と解決機構に関しては次章で説明する。

4.3.1 イベントの転送

提案する有線接続システムは、Bluetooth 間の通信において発生する HCI イベントの転送を行うことで透過な有線接続を実現する。Bluetooth の Baseband 層および LM 層で発生したイベントは、HCI を介して HCI イベントとして上位層へ通知される。このような HCI イベントのなかで、特に接続要求などの通信リンクに関わる HCI イベントは、接続相手の Bluetooth 機器が HCI コマンドを実行によって発生する。つまり、HCI イベントには、そのイベントを発生させる対となる HCI コマンドが存在する。この関係を利用して提案システムでは、Manager が Bluetooth インターフェースを介して HCI イベントの通知を受信するとリモート側の Manager に対して HCI イベントを受信したことを有線伝送路を通して通知し、受信したイベントの対となる HCI コマンドをリモート側の Bluetooth 機器で実行することでイベントの転送を実現する。図 4.2 に HCI コマンドによる HCI イベントの転送を図示する。

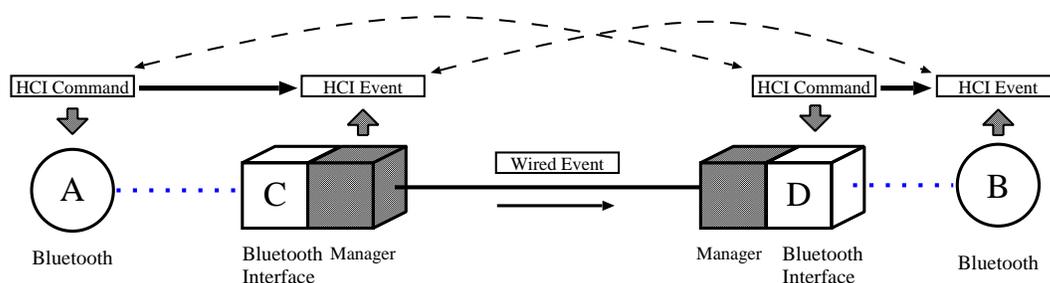


図 4.2: HCI コマンドによる HCI イベントの転送

イベントの転送の実例を、接続要求イベントの転送を用いて説明する。図 4.3 にローカル側の Bluetooth インターフェイスが Bluetooth から接続要求を受信した際のローカル側の Manager の動作を示す。Bluetooth A から接続要求を受信した Bluetooth Interface-L は、Manager-L に対して、HCI Connection Request event を通知する。このイベントを受けた Manager-L は、リモート側の Manager-R に対してリモート側の Bluetooth B に対しての接続要求受け取ったことを通知する Wire_Connection_Request を送信する。図 4.4 にローカル側 Manager-L からの Wire_Connection_Request を受けた際のリモート側の Manager-R の動作を示す。Wire_Connection_Request を受信した Manager-R は、リモート側の Bluetooth B への接続を要求する HCI Create Connection Request コマンドを Bluetooth Interface-R へ送信する。これによって、Bluetooth-L からの接続要求をリモート側の Bluetooth-R に転送することができる。

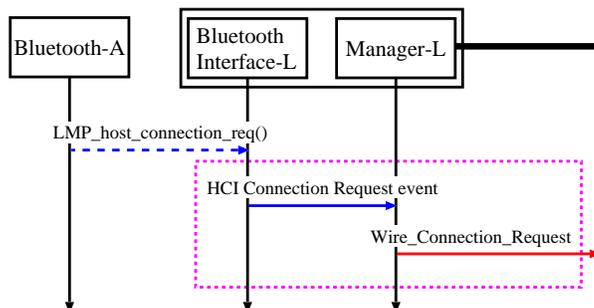


図 4.3: ローカル側における接続要求の受信

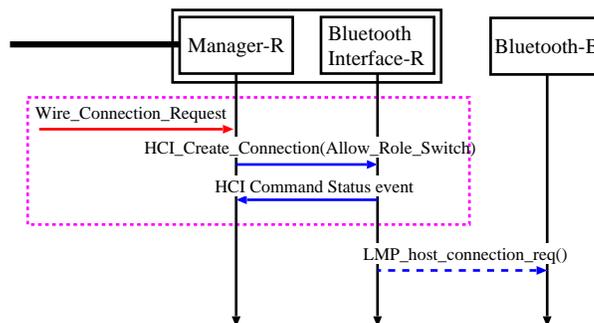


図 4.4: ローカル側からの接続要求の受信

このように Manager が Bluetooth インターフェイスからの HCI イベントの解釈し、有線伝送を通じてリモート側の Manager に対して要求を出すことでリンクの接続要求、切断要求などを転送し、Bluetooth 機器間に存在する有線接続システムを透過にする。Manager が Bluetooth インターフェイスから受信した際にリモート側へ通知しなくてはならない HCI イベントを表 4.1 に示す。

転送を必要とする HCI イベント	通知される内容
Connection Complete event	通信リンクの確立
Connection Request event	通信リンクの接続要求
Disconnection Complete event	通信リンクの切断
Encryption Change event	暗号化モードの変更 (暗号化の通知)
QoS Setup event	通信リンクの QoS パラメータの設定完了
Role Change event	Bluetooth 役割の変更
Mode Change event	Bluetooth 通信モードの変更
PIN Code Request event	PIN コードの要求 (認証の通知)
Link Key Request event	リンクキーの要求 (認証の通知)
Max Slots Change event	通信リンクで用いる最大スロット数
Connection Packet Type Changed event	通信リンクで用いるパケットタイプの変更

表 4.1: 転送を必要とする HCI イベント

4.3.2 データパケットの転送

提案する有線接続システムは、Bluetooth 間の通信において発生する HCI データパケットの転送を行うことで透過な有線接続を実現する。接続された Bluetooth 機器からの通信データは、Bluetooth インターフェースを介して Manager に HCI データパケットとして渡され、この HCI データパケットを有線伝送路を用いて転送する。HCI では、通信リンクの識別をリンク接続時に設定されるコネクション・ハンドルを用いて管理されており、有線伝送路を介して中継された HCI データパケットは、データパケットの送信先を対応する通信リンクのコネクション・ハンドルに置き換えて Bluetooth インターフェースに送信されることで、ローカル側の Bluetooth 機器からの HCI データパケットをリモート側へ届けることができる。図 4.5 にデータパケットの転送時の動作を示す。

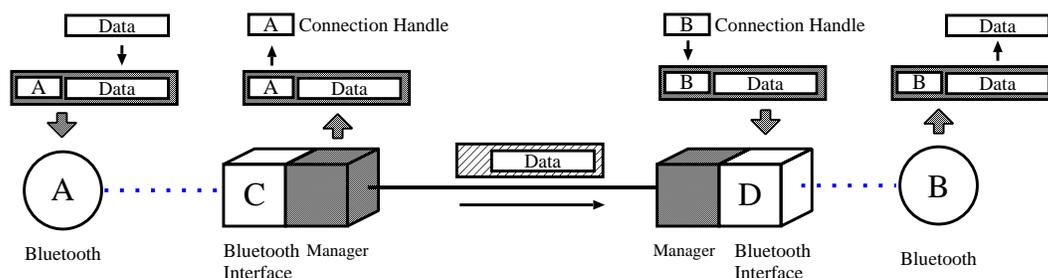


図 4.5: データパケットの転送

4.4 提案システムのプロトコルスタック

本研究で提案する有線接続システムと Bluetooth 機器のプロトコルスタックの関係を図 4.6 に示す。HCI における HCI イベントおよび HCI データフレームを転送する提案システムは、L2CAP 層を含む上位層からは透過な有線接続システムとなっていることがわかる。

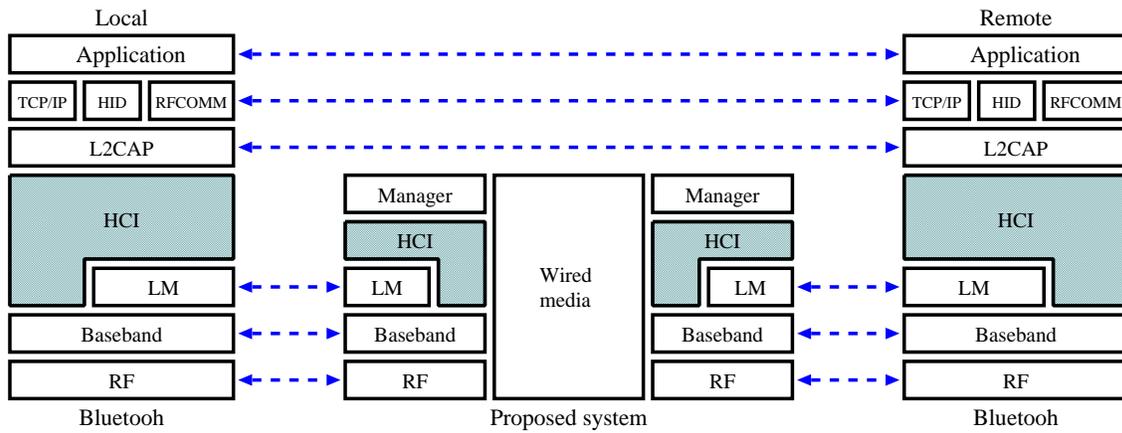


図 4.6: 有線接続システムのプロトコルスタック

第5章 提案システムの課題と解決手法

本章では, 提案システムによって Bluetooth 機器間を有線接続する際の課題とその解決のために提案システムが用いる解決手法を説明する.

5.1 ピコネット内同期

本研究で提案する有線接続システムでは, ローカル側, リモート側でそれぞれ異なるピコネットを形成し, Bluetooth 機器間の相互接続を実現する. そのため, Bluetooth 機器間で論理リンクを確立し通信を行うためには, 有線接続システムのローカル側とリモート側それぞれにおいて, 接続される Bluetooth 機器と有線接続システムの Bluetooth インターフェイス間で, 周波数軸および時間軸を同期するピコネット内同期が確立されている必要がある. しかし, 同期確立のためのフェーズである問い合わせフェーズおよび呼び出しフェーズは Bluetooth 機器と有線接続システムの Bluetooth インターフェイスの Baseband 層で閉じた形で実行されるため, これを検知することはできない. このため, ローカル側でピコネット内同期が完了し, 接続要求が発生した時点では, リモート側でピコネット内同期が完了していないことから, 接続要求発生後に問い合わせフェーズおよび呼び出しフェーズを実行することになる. しかし, 問い合わせフェーズだけでも最低 10

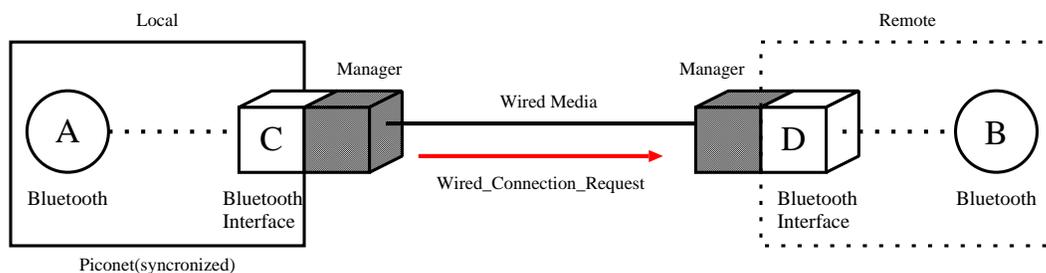


図 5.1: ピコネット内同期問題

.24sec[1] 必要とするため, 接続タイムアウトなどの問題が発生する可能性がある. これに対して, 本研究では有線接続システムが投機的に問い合わせフェーズを実行し, 接続対象となる Bluetooth 機器の Bluetooth デバイスアドレス, クロックオフセットを取得することで, 問い合わせフェーズの省略, 呼び出しフェーズの高速化を行い, 問題の解決を図る. この機構を用いない場合における Bluetooth 機器と有線接続システム間のメッセー

ジシーケンス図を図 5.2 に、用いた場合のメッセージシーケンス図を図 5.3 に示す。問い合わせフェーズを予め行うことで、接続時間が短縮されていることがわかる。

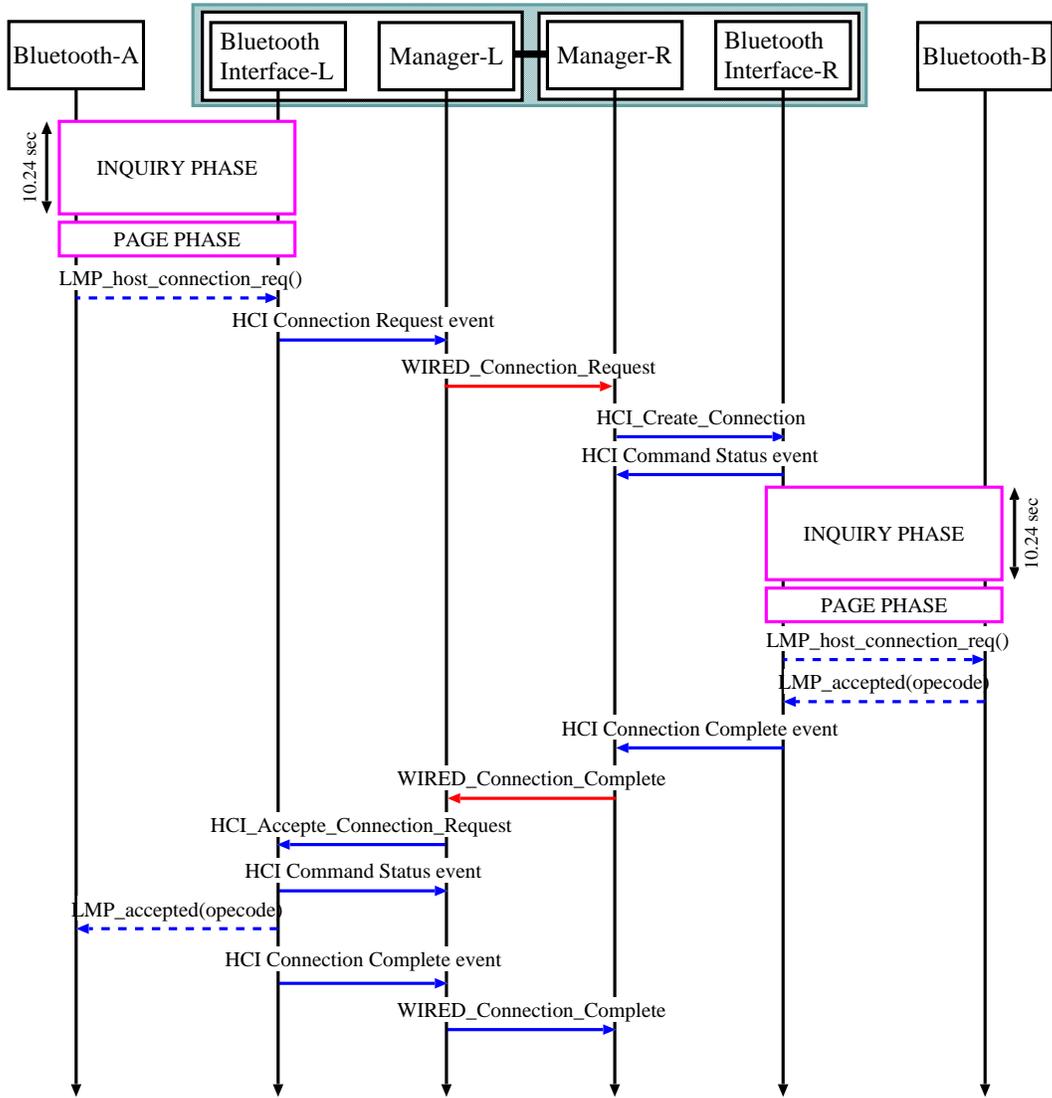


図 5.2: ピコネット内同期問題

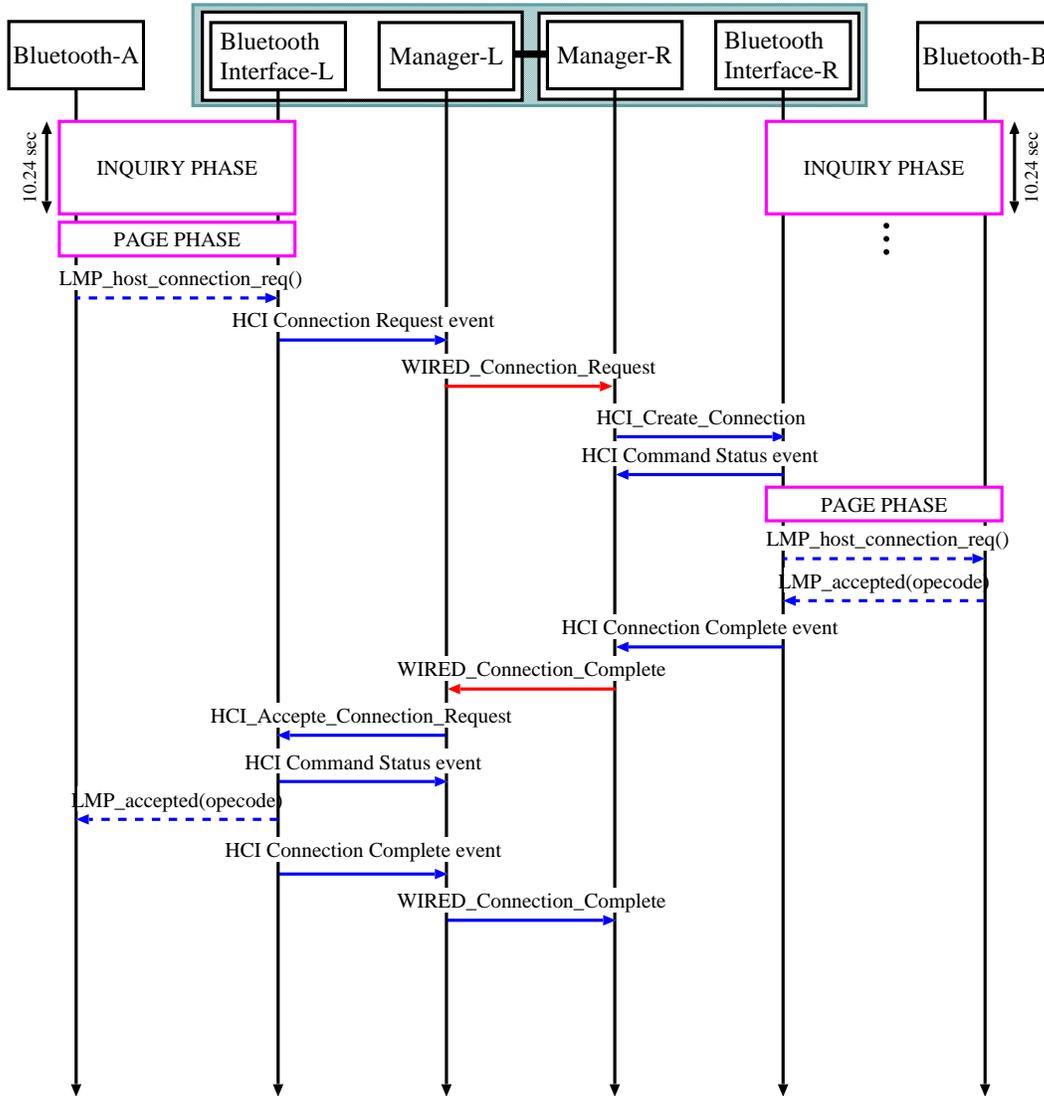


図 5.3: ピコネット内同期問題の解決機構

5.2 リモートの Bluetooth 機器情報の取得

一般的な Bluetooth アプリケーションは、接続相手の選択、接続を次の手順で行う。

1. 周辺の Bluetooth 機器のスキャン
2. 発見された機器の名前、種類、サービス情報の取得
3. 取得された情報を元に接続相手を選択し接続。

接続相手を判断するための情報となる機器の名前、種類、サービスなどの情報は LM 層におけるリンク・マネージャで管理され、Bluetooth 機器はこれらの情報に関する問い合わせを受けた場合には、LMP(Link Management Protocol) を用いて返答する。このため、有線接続システムを用いた透過な機器間の接続を実現するためには、有線接続システムは、ローカル側の Bluetooth 機器に対してリモート側の Bluetooth 機器の情報を提供できなくてはならない。しかし、有線接続システムを用いた場合、有線接続システムが持つ各 Bluetooth インターフェースのリンク・マネージャは、初期状態では提供すべき情報を持っていない。

これに対して、本研究では有線接続システムが Inquiry, HCI_Remote_Name_Request を行うことでリモート側の Bluetooth 機器情報を集め、取得した情報をローカル側へ送信する。これにより、ローカルの Manager は、リモートの Bluetooth 機器の種類を表す CoD(Class of Device) と Bluetooth 機器のユーザフレンドリーな名前を取得することができる。リモート側で取得した情報をローカル側の Bluetooth インターフェースへ HCI コマンド HCI_Write_Class_of_Device, HCI_Write_Local_Name を用いて反映することで、ローカルの Bluetooth 機器に対して、リモートの Bluetooth 機器の情報を提供する。この機構によって、有線接続システムのもつローカル側の Bluetooth インターフェースの情報をリモート側の Bluetooth 機器の情報と一致させ、接続対象となる Bluetooth 機器からは有線接続システムの Bluetooth インターフェースをリモート側の機器であるかのように見せることができる。

5.3 Bluetooth デバイスアドレス

Bluetooth 規格では、同一ピコネット内における周波数ホッピングパターンをマスターとなる Bluetooth 機器の Bluetooth デバイスアドレスから算出するため、1つの Bluetooth インターフェースには必ず1つの Bluetooth デバイスアドレスが必要である。これは、Ethernet ブリッジのようにブリッジのインターフェース自体はアドレスを持たず、接続された機器のアドレスをそのまま通すシステムを Bluetooth では実現できないことを意味している。

また、Bluetooth では、接続相手の識別を Bluetooth デバイスアドレスをもとに行うため、リモート側に複数の Bluetooth 機器が存在する場合には、ローカル側には、複数の

Bluetooth デバイスアドレスが必要となる．このため，本研究で提案する有線接続システムでは複数の Bluetooth インターフェースを有線接続システムが持つことで，接続される Bluetooth 機器の識別を可能にする（図 5.4）．これに伴い有線接続システムの Manager では，複数の Bluetooth インターフェースと接続される Bluetooth 機器の接続リンクを管理する機構が必要となり，接続テーブルを作成し，これを管理する．

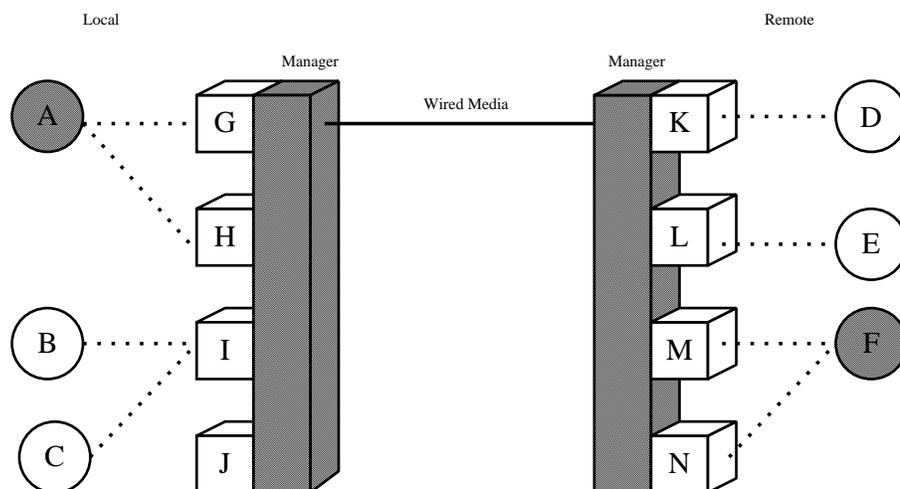


図 5.4: 有線接続システムの接続の仕組み

5.4 Bluetooth インターフェースの MTU(Max Transfer Unit)

提案する有線接続システムでは，Bluetooth インターフェースとして，パッケージ化された Bluetooth モジュールを利用する．Bluetooth モジュールは，様々なインターフェースのものが存在するため各インターフェースのもつ MTU がそれぞれ異なる．このためローカル側からの HCI データパケットがリモート側の Bluetooth インターフェースのもつ MTU よりも大きい場合があり，この場合 HCI データパケットのコネクション・ハンドルを置き換えただけでは，リモート側の Bluetooth インターフェースに送ることができない．提案システムでは，Bluetooth インターフェースの MTU よりも大きい HCI データパケットを受け取ったりリモート側の Manager は送信先 Bluetooth インターフェースの MTU に分割し，Bluetooth に送ることで，MTU の異なる Bluetooth インターフェース間の通信を可能にする．HCI ACL データ・パケットには，ヘッダ部に L2CAP パケットの始まりを表す Packet_Boundary_Flag が存在する．このため分割の際に HCI ACL データ・パケットに L2CAP パケットの始まりを示すフラグ “10(2進)” が設定されている場合，分割後の HCI ACL データパケットは，1 番目のパケットの Packet_Boundary_Flag に “10”

をセットし、後半のパケットには L2CAP パケットの始まりではないことを表す “01” をセットする。

5.5 ローカル側リンクとリモート側リンクで使用するパケットタイプの一致

提案する有線接続システムでは、Bluetooth インターフェースからの HCI イベントをもとに対応する HCI コマンドを利用することで、接続される Bluetooth 機器間の透過接続を実現する。このため、Baseband 層および LM 層から HCI イベントを介して通知されない情報に関しては利用することができない。接続されている通信リンクに利用しているパケットタイプは、LM 層において決定され HCI イベントから得ることができない情報の 1 つである。利用されているパケットタイプを得ることができないことから、ローカル側リンク、リモート側リンクで異なるパケットタイプを利用する可能性がある。ローカル側とリモート側でそれぞれ設定される通信リンクで異なるパケットタイプが利用された場合、一方の通信リンクが一方の通信リンクよりも早くデータパケットを送信することが可能になる。これによって、図 5.5 のように通信速度の差分のデータパケットが提案システムでバッファリングされることになり、バッファあふれを起こす可能性がある。これを回避するために、双方のリンクで用いられるパケットタイプを一致させることが必要となる。

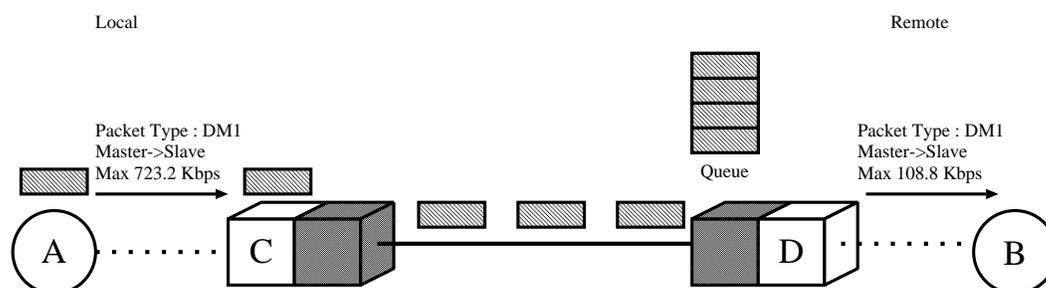


図 5.5: パケットタイプの相違

この問題に対して、HCLChange_Paket_Type コマンドを利用して、予め最も通信速度の低いパケットタイプ DM1 を利用するように提案システム側から設定する手法と、キューにバッファリングされるデータパケットが多くなった時点で、データパケットの通信量から通信リンクのパケットタイプを推定し、パケットタイプの変更を加える手法が考えられる。いずれの手法においても、通信量の低いパケットタイプで設定されている通信リンクを通信量の高いパケットタイプに変更することは困難であることから、通信量の低いパケットタイプに変更する必要がある。

5.6 有線伝送路でのデータの保証

提案システムが用いる有線伝送路にはデータの保証が必ず必要である。ACL リンクでは、Baseband 層において再送制御を行うことでデータの保証を行うため、ACL リンクを利用する上位層はデータの消失を前提としていない。このため、提案システムを用いる場合は、有線伝送路上でデータの消失は許されない。提案システムを用いる際には、有線伝送路の通信メディアにデータの保証が可能な通信メディアを用いるか、保証されない場合は上位層でその保証を行う必要がある。また、SCO リンクに関しては、データの保証は必要ではない。しかし SCO リンクは帯域保証型の通信リンクであるため、有線伝送路には帯域保証が必要である。また、SCO リンクは音声などのリアルタイム性の高いデータを伝送するために用いられるため、SCO データパケットの有線伝送路上での転送にはジッタの少ない伝送路が望まれる。

5.7 接続認証

Bluetooth は、未認証端末間の誤接続を防止するための接続認証機能をもつ。接続認証を行う際には、ペアリングを行うための接続認証コードとして共通の PIN を用い、同じ PIN をもつ Bluetooth 機器同士が接続可能となる。このため、ローカル側の Bluetooth 機器が提案システムに接続する際に接続認証を要求した場合、リモート側の Bluetooth 機器の PIN を提案システムが保有していなくてはならない。しかし、PIN はセキュリティに関わるパラメータであることから、接続対象となる Bluetooth 機器の PIN を提案システムが動的に取得することはできない。

この問題に対して、本研究では有線接続システムに接続対象となる Bluetooth 機器の PIN を予め手動で入力し固定することで接続認証に関する問題の解決を図る。提案システムは、有線伝送路を利用する接続システムであることから、固定設置される機器であると想定される。このため、PIN に関しても毎回入力するわけではなく固定の PIN を持たせること、または接続される Bluetooth 機器のアドレスと PIN のセットを入力し、データベースとして管理することでこの問題の解決を図る。また、ペアリング時に生成されるリンク・キーも同様に管理することで、一度ペアリングを行った Bluetooth 機器と再び認証を行う場合におけるリンク・キーを利用した認証を可能にする。

5.8 接続セットアップ時のセキュリティ・モードの一致

本研究で提案する有線接続システムでは、ローカル側、リモート側でそれぞれ異なるピコネットを形成し、Bluetooth 機器間の相互接続を実現する。Baseband 層、LM 層における通信リンクレベルで認証、暗号化を行う Bluetooth を有線接続する提案システムでは、ローカル側、リモート側それぞれで認証、暗号化を設定をする必要がある。Bluetooth 機器間において、通信リンクを確立する際には、セキュリティの設定が高い Bluetooth 機器に

合わせてリンクのセキュリティ・モードが設定される。この機構を別々に通信リンクを設定する提案システムにおいても実現できなくてはならない。この問題の解決には、単純なイベントの転送とは異なるイベントの転送処理が必要となる。

このため、提案システムがローカル側、リモート側の機器が ACL リンク確立のセットアップ時にセキュリティ・モードを要求した場合に、ローカル側が要求した場合、リモート側が要求した場合のそれぞれのパターンに関して提案システム内でセキュリティ・モードの一致を図るための処理を施す。例えば、リモート側でセットアップ時に認証・暗号化を要求された場合には、ローカル側においてもセットアップ時に認証・暗号化を行うように Bluetooth インターフェイスを設定するなどである。具体的なセキュリティ・モードの一致に関する処理の詳細は、次章で ACL 確立時のメッセージ・シーケンス・チャートを示す。

第6章 メッセージ・シーケンス・チャート

本章では、提案提案する有線接続システムと接続機器間のイベントの流れをメッセージ・シーケンス・チャートを用いて表す。メッセージ・シーケンス・チャートを表す際、簡素化のため、Bluetooth 機器と Bluetooth インターフェース間でやりとりされる LMP メッセージは最小限のメッセージのみを表記し、また、インターフェース間の転送エラーなどは考慮しないものとしてある。

6.1 Bluetooth デバイス情報の取得

6.1.1 情報取得

図 6.1 に情報取得時のメッセージ・シーケンス・チャートを示す。情報取得フェーズは、有線接続システムが起動する際に、他の有線接続システムへ周辺の Bluetooth 機器の情報を与えるために周辺の Bluetooth 機器を検索し、それぞれの名前を取得するためのフェーズである。具体的な手順は、まず Inquiry を行い、周辺の機器の Bluetooth デバイスアドレス、および CoD を取得する。次に、各 Bluetooth 機器の名前を取得するために HCI_Remote_Name_Request を用いてそれぞれの Bluetooth 機器の名前を取得する。提案システムは、これらの情報をまとめ他の提案システムへ送信する。また、このとき取得されて他の情報は、ACL リンク接続時のパラメータとして利用するため、提案システムに保存する。

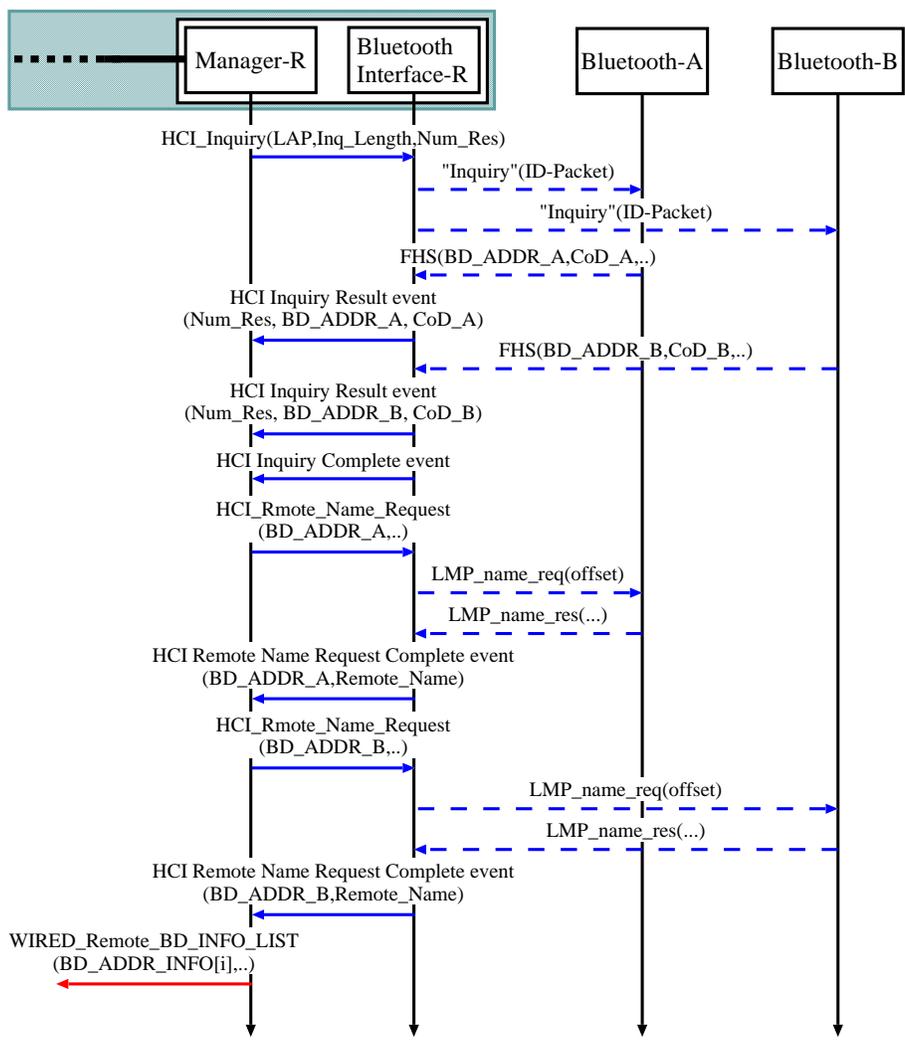


図 6.1: Bluetooth デバイス情報の取得

6.1.2 Bluetooth デバイス情報の反映

図 6.2 にデバイス情報反映フェーズのメッセージ・シーケンス・チャートを示す。情報反映フェーズは、他の Bluetooth 機器から得られたリモートの Bluetooth 機器の情報を自身の持つマネージャへ登録する。得られる情報はリモート側に存在する Bluetooth 機器の、Bluetooth デバイスアドレス、CoD, 名前であり、CoD は HCI_Write_Class_of_Device コマンド、名前は、HCI_Change_Local_Name コマンドを用いることで提案システムの Bluetooth インターフェースに反映する。これにより、各 Bluetooth インターフェースに対して、inquiry や名前のリクエストがなされた場合に、リモート側の Bluetooth 機器の情報を提供することができる。

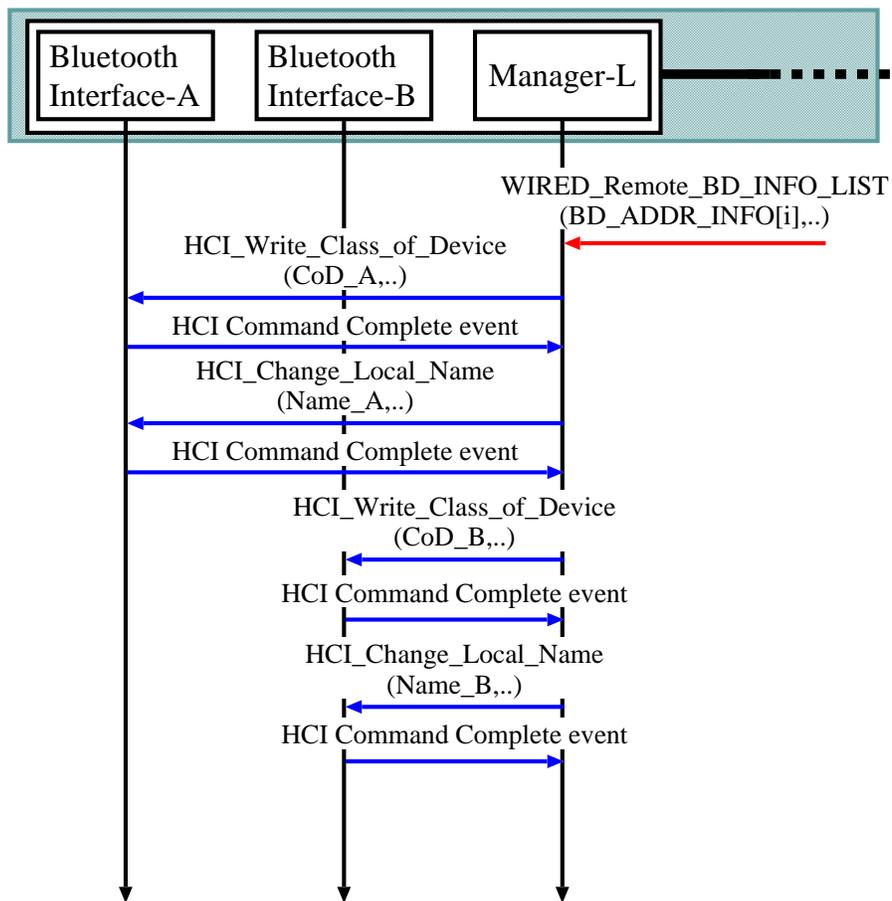


図 6.2: Bluetooth デバイス情報の反映

6.2 ACL 接続要求フェーズ

接続要求フェーズを図 6.3, 図 6.4, 図 6.5 に示す。提案システムの Bluetooth インターフェースは, 初期状態ではコネクション・セットアップ時のセキュリティに関するモードをすべて DISABLE である。提案システムは, Bluetooth インターフェース側からコネクション・セットアップ時の認証・暗号を要求することはなく, 接続される機器にあわせてセキュリティ・モードを動的に変化させる。図 6.3, 図 6.4, 図 6.5 に ACL 接続フェーズを 3 つのサブシナリオに分けて示す。

サブシナリオ 1 Bluetooth-B が ACL 接続要求を拒否した場合

Bluetooth-B が ACL 接続要求を拒否した場合, リモート側の Bluetooth Interface-R から Manager-R には HCI Connection Complete イベントが戻される。このとき, Status には, Bluetooth-B が ACL 接続要求を拒否した場合の理由 Reason パラメータが戻され, これを受けた Manager-R が WIRED_Connection_Complete に Status を入れてローカル側へ送信する。WIRED_Connection_Complete の Status が Host Reject であった場合, Manager-L が Bluetooth-A に対して Status に Reason パラメータに代入し, Host_Reject_Connection_Request コマンドを実行することで Bluetooth-A からの ACL 接続要求を拒否する。

サブシナリオ 2 Bluetooth-B が ACL 接続要求を受け入れた場合

Bluetooth-B が ACL 接続要求を受け入れた場合, HCI Connection Complete イベントの Status には Success が戻る。WIRED_Connection_Complete の Status から, Manager-L は HCI_Accept_Connection_Request コマンドを実行し, コネクションを受け入れる。Bluetooth 機器間の直接接続の場合は, ペアリング・認証・暗号化のセットアップフェーズへと移るが, 提案システムでは Bluetooth-A および Bluetooth-B の要求するセキュリティ・モードによって動作が異なる。サブシナリオ 2 は, Bluetooth-A, Bluetooth-B がどちらもセキュリティを要求しなかった場合のメッセージ・シーケンス・チャートである。また, ローカル側, リモート側のそれぞれのリンクに対するコネクション・ハンドル (または同等のもの) もそれぞれの Manager には通知しなくてはならない。このコネクションハンドルを用いて, Manager は対となるリンクの識別を行うためである。

サブシナリオ 3 Bluetooth-B が役割変更付き ACL 接続要求を受け入れた場合

Bluetooth-B が役割変更を要求した場合, Manager-R には, HCI Role Change イベントが戻る。Manager-R は, WIRED_Role_Change を送信しリモート側の Bluetooth-B が役割変更を要求していることをローカル側へ伝える。この情報をもとに, Manager-L が HCI_Accept_Connection_Request コマンドを実行する際に役割変更を要求することでローカル側の Bluetooth-A とリモート側の Bluetooth-B の役割を直接接続した場合と一致させる。

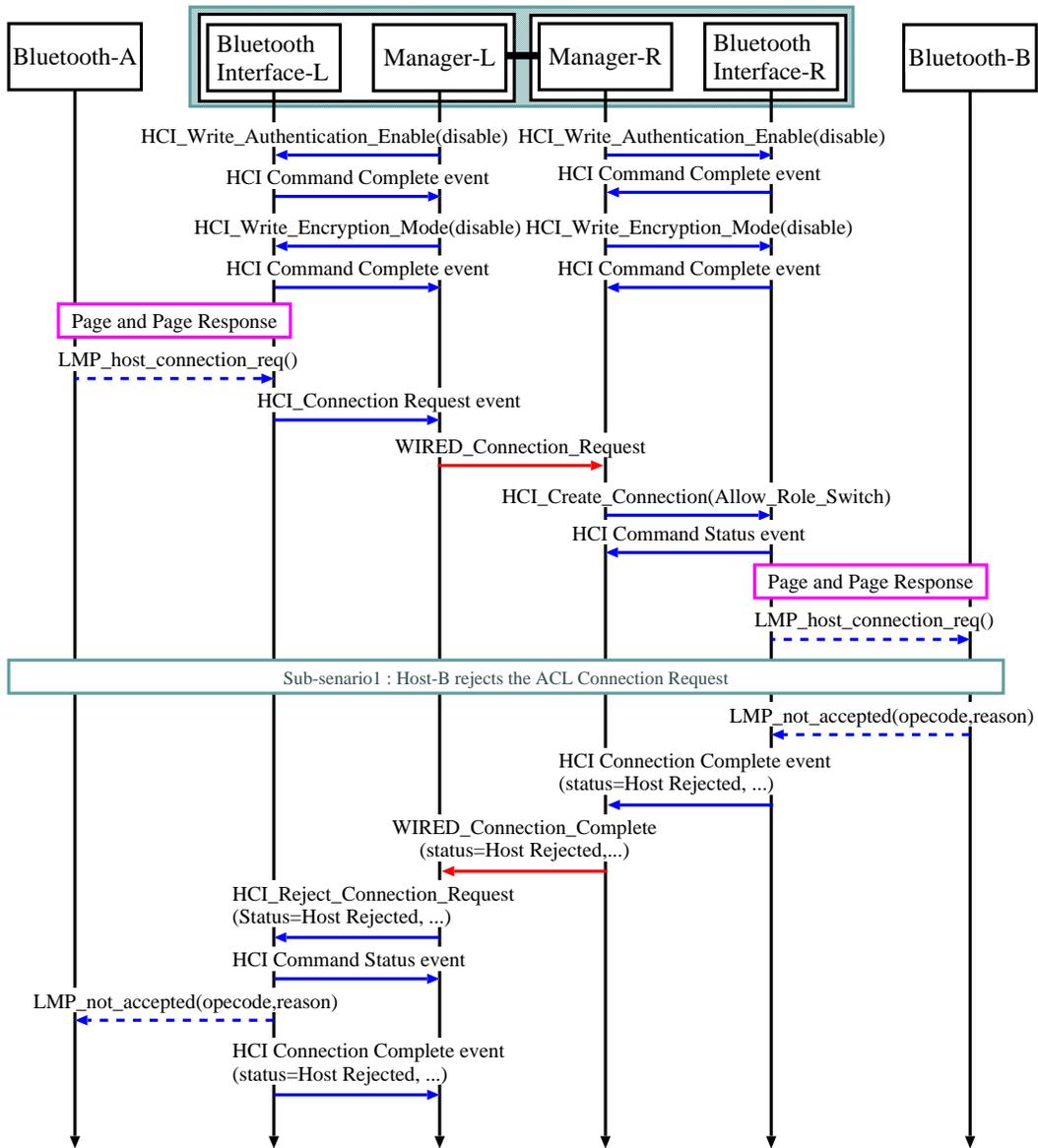


図 6.3: ACL 接続要求フェーズ 1

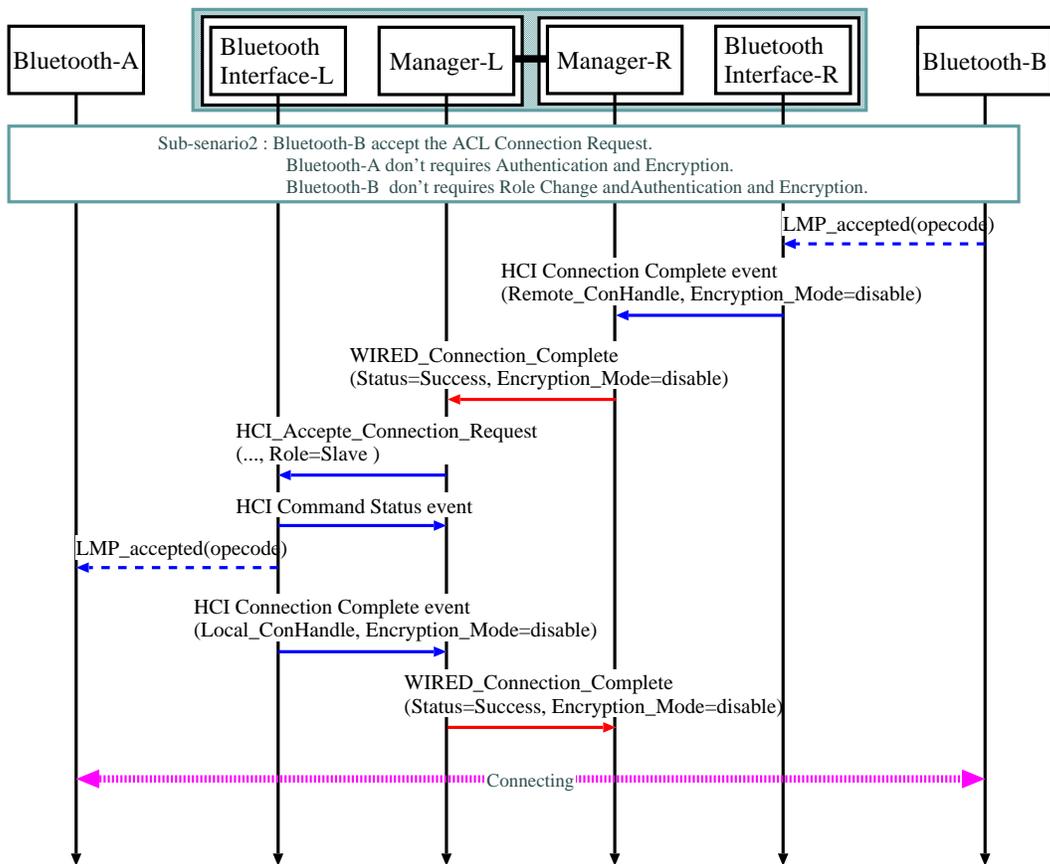


図 6.4: ACL 接続要求フェーズ 2

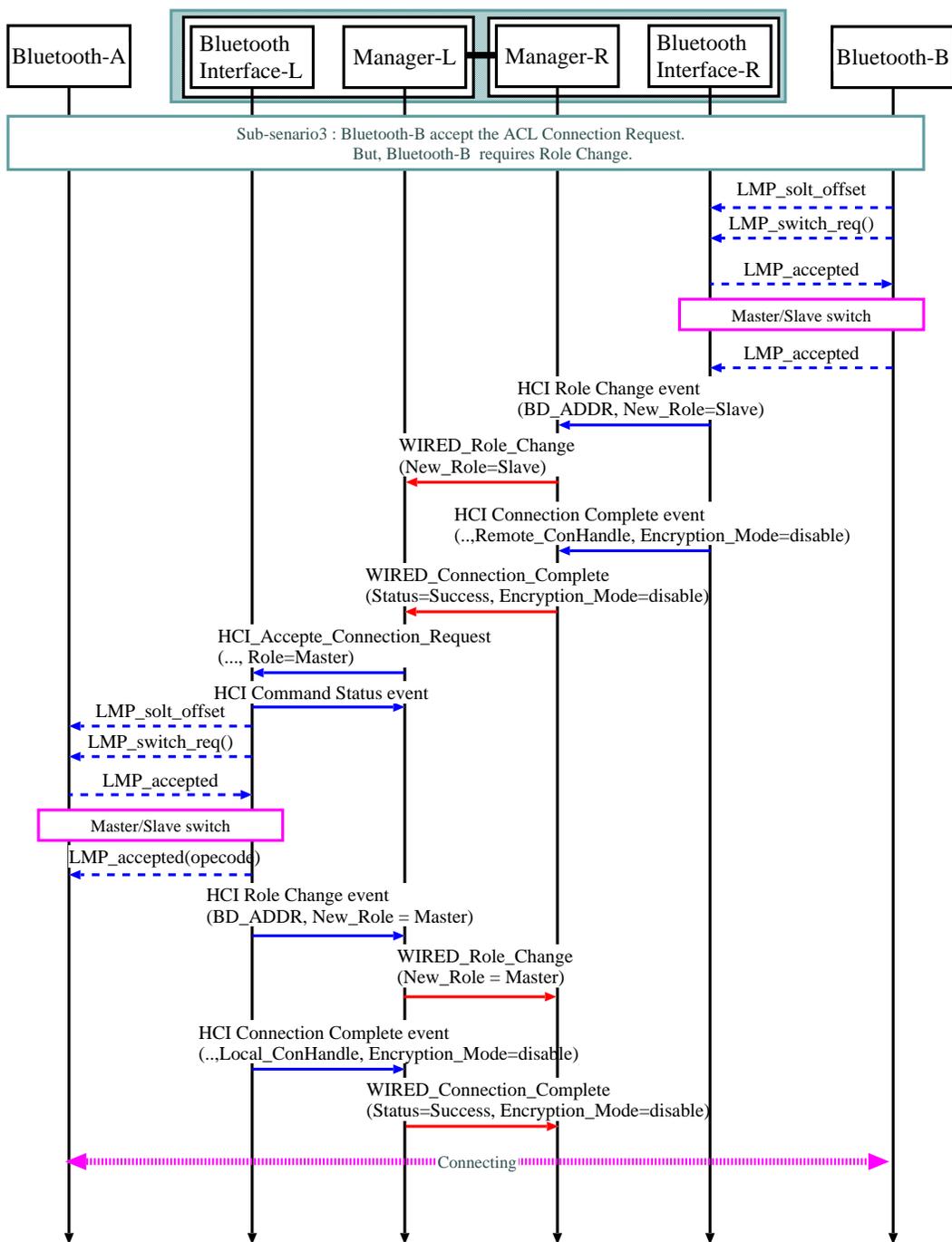


図 6.5: ACL 接続要求フェーズ 3

6.3 認証および暗号化

ACL 確立時に、接続される Bluetooth 機器からセキュリティ・モードに関する要求があった場合、提案システムは、ローカル側とリモート側のリンクにおいてリンク確立時のセキュリティ・モードを一致させる必要がある。提案システムでは、リモート側の Bluetooth 機器が接続セットアップ時にセキュリティ・モードを要求する場合と、ローカル側の機器が要求する場合で、それぞれ異なる手順でセキュリティ・モードの一致を図らなくてはならない。

リモート側の Bluetooth 機器がペアリング・認証を要求した場合

図 6.6 にリモートの Bluetooth-B がペアリング・認証を要求した場合のメッセージ・シーケンス・チャートを示す。リモート側の Bluetooth-B が ACL コネクション確立時にペアリングを要求した場合、リモート側の Manager-R には、HCI Link Key Request または、HCI PIN Code Request イベントが戻る。これらのイベントを受けた Manager-R はローカル側の Manager-L に対して、認証要求を受けたことを通知する。図 6.6 では、HCI PIN Code Request イベントに対して、WIRED_PIN_Code_Request を送信している。Manager-L はこの通知を受け、Bluetooth Interface-L に対して、HCI_Authentication_Enable コマンドを利用し、ローカル側の ACL 確立時における認証を設定する。これによって、ローカル側で WIRED_Connection_Complete を受け HCI_Accept_Connection_Request を実行すると、ローカル側における ACL 確立のセットアップ時にも認証を行うことになり、セキュリティ・モードを一致させることができる。また、ローカル側、リモート側の両方の Bluetooth 機器が認証を要求した場合もメッセージ・シーケンスは同様である。

リモート側の Bluetooth 機器が暗号化を要求した場合

図 6.7 にリモート側の Bluetooth 機器が暗号化を要求した場合におけるメッセージ・シーケンス・チャートを示す。リモート側の Bluetooth-B が暗号化を要求した場合には、リモート側の Manager-R への HCI Connection Complete イベントのパラメータ Encryption_mode に暗号化のモードが戻される。ローカル側の Manager-L への WIRED_Connection_Complete へ暗号化のモードも通知することで、Manager-L は HCI_Encryption_Mode コマンドを実行し、ローカル側 ACL 確立時における暗号化を設定する。これによって、ローカル側、リモート側それぞれの ACL のセキュリティ・モードを一致させることができる。また、ローカル側、リモート側の両方の Bluetooth 機器が暗号化を要求した場合もメッセージ・シーケンスは同様である。

ローカル側の Bluetooth 機器が認証を要求した場合)

図 6.8 にリモート側の Bluetooth 機器がなにもセキュリティを要求せず、ローカル側の Bluetooth 機器が認証を要求した場合におけるメッセージ・シーケンス・チャートを示す。ローカル側の Bluetooth 機器のセキュリティ・モードを提案システムが知る方法は、リモート側の Bluetooth 機器の場合と同様に ACL 確立セットアップ時

のイベントである。しかし、これらセキュリティ・モードのイベントは、ローカル側では、HCI_Accept_Connection_Request コマンドを実行するまで得ることができないため、リモート側での ACL 確立後にローカル側のセキュリティ・モードを得ることになる。このため、提案システムではリモート側において ACL 確立時にセキュリティが要求されず、ローカル側においてセキュリティ・モードの要求を受けた際は、一度リモート側 ACL の切断を行う。その後、ローカル側の要求に合わせてリモート側の Bluetooth インターフェースを設定し直し、再接続することでセキュリティ・モードの一致を図る。

ローカル側の Bluetooth 機器が暗号化を要求した場合

図 6.9 にリモート側の Bluetooth 機器がなにもセキュリティを要求せず、ローカル側の Bluetooth 機器が暗号化を要求した場合におけるメッセージ・シーケンス・チャートを示す。この場合も、同様にリモート側 ACL の切断、セキュリティ・モードの再設定を行うことで、ローカル側、リモート側の ACL 確立セットアップ時のセキュリティ・モードの一致を図る。

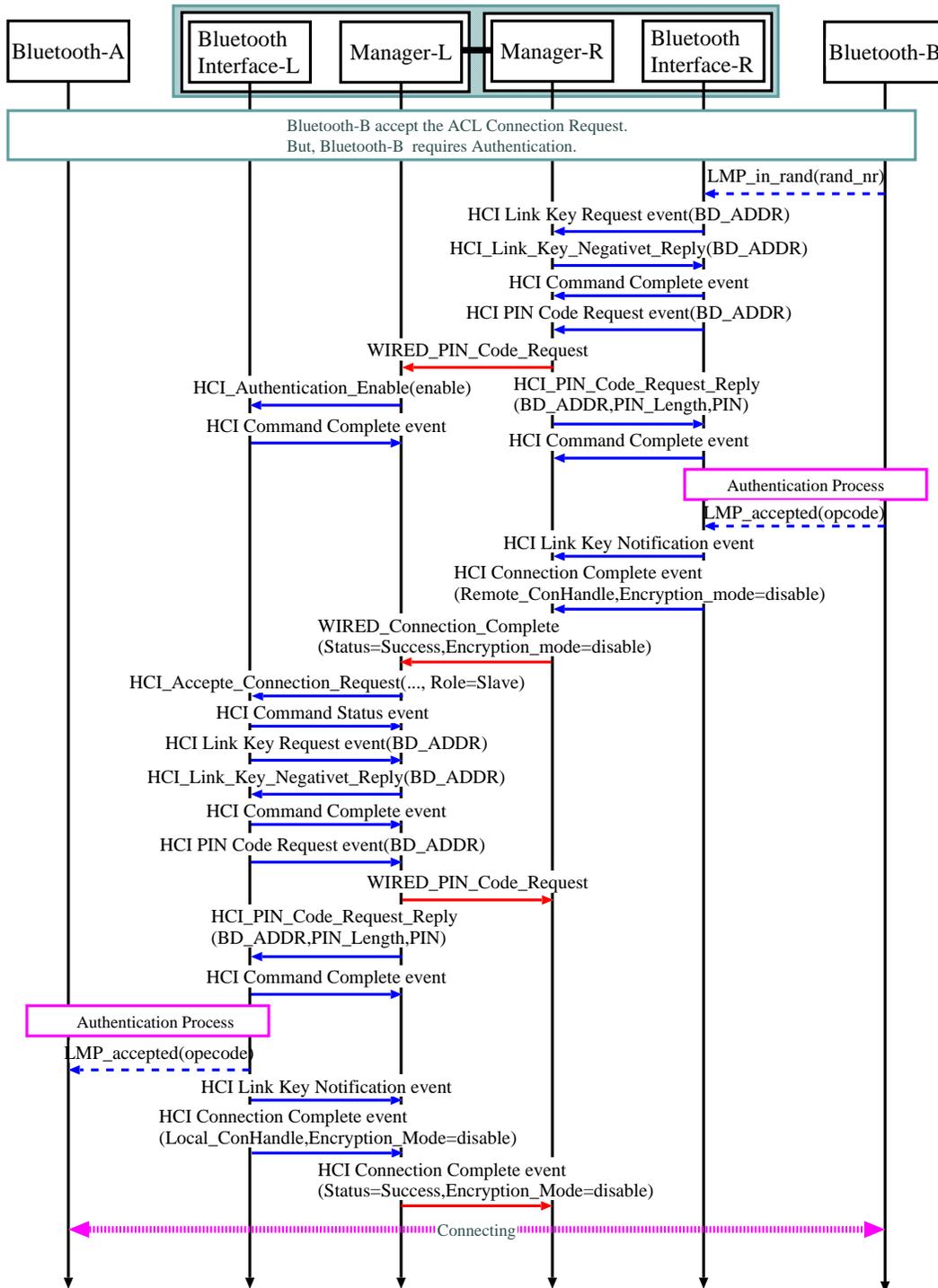


図 6.6: リモート側の Bluetooth 機器がペアリング・認証を要求

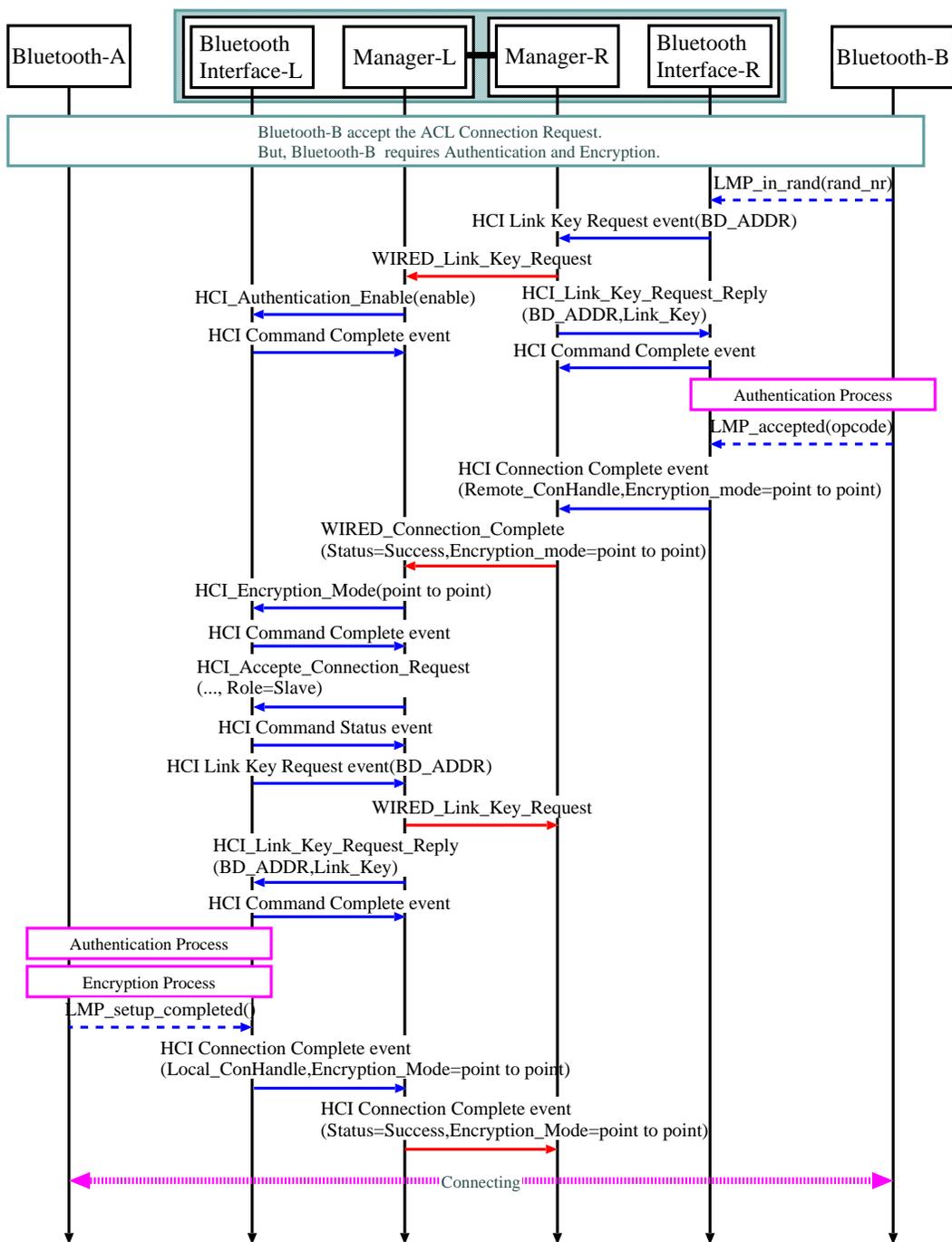


図 6.7: リモート側の Bluetooth 機器が暗号化を要求

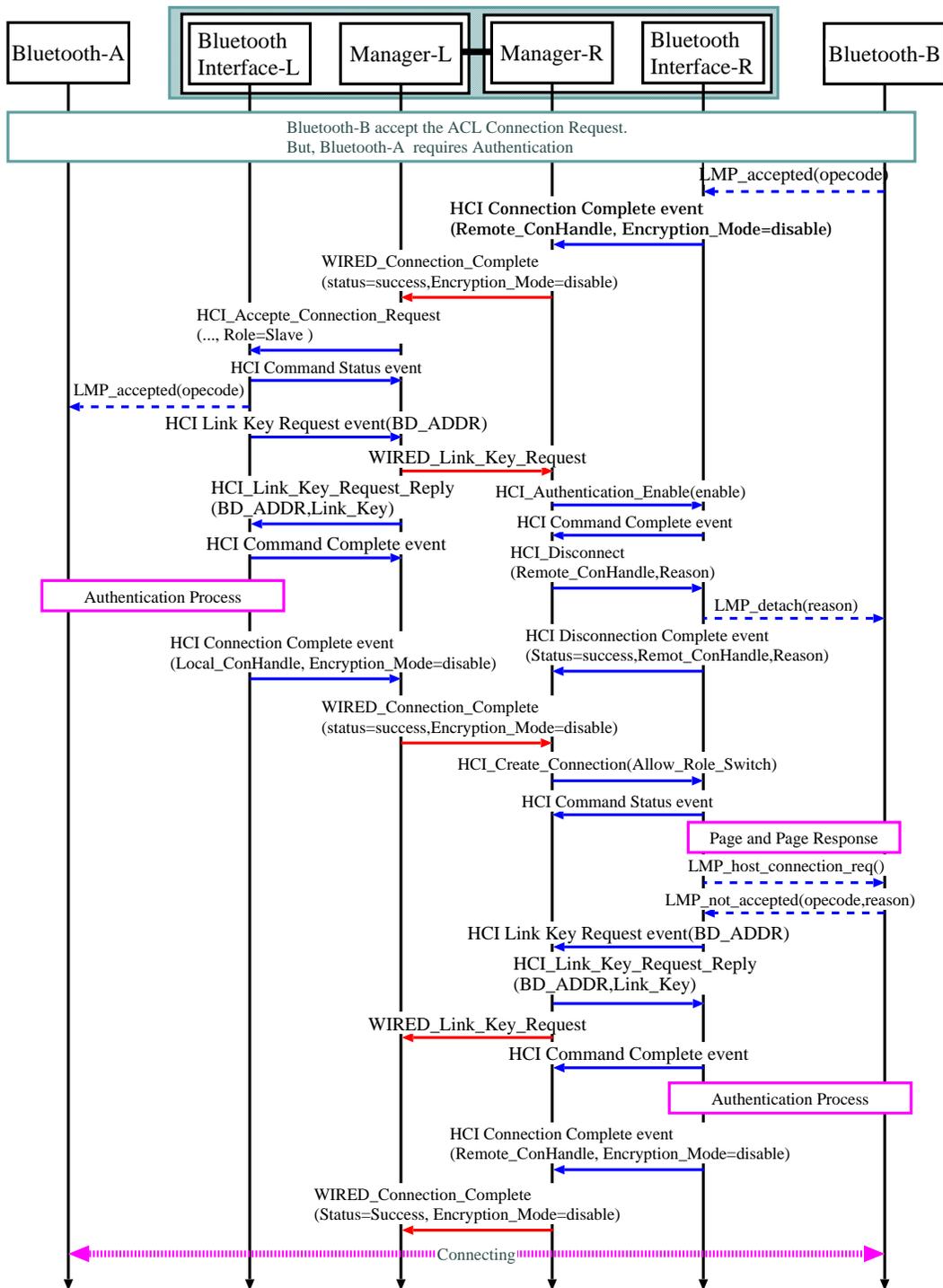


図 6.8: ローカル側の Bluetooth 機器が認証を要求

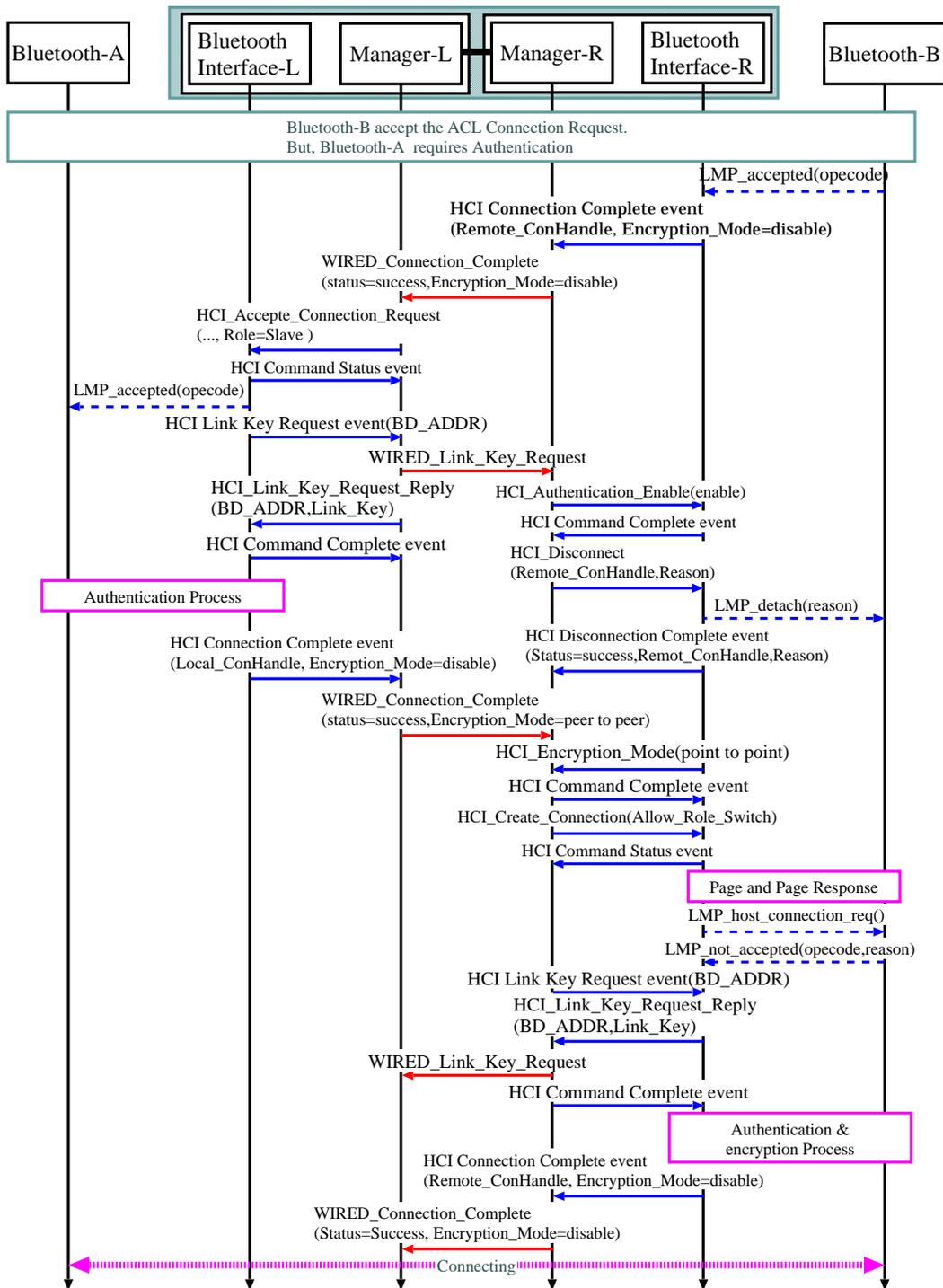


図 6.9: ローカル側の Bluetooth 機器が暗号化を要求

6.3.1 ACL 切断

図 6.10 に ACL 切断時のメッセージ・シーケンス・チャートを示す。片方の ACL が切断された場合、Manager へは HCI Disconnection Complete イベントが戻る。このイベントに対して、WIRED_Disconnection_Complete を送信し、HCI_Disconnect コマンドを実行することでイベントの転送を実現し ACL を切断する。

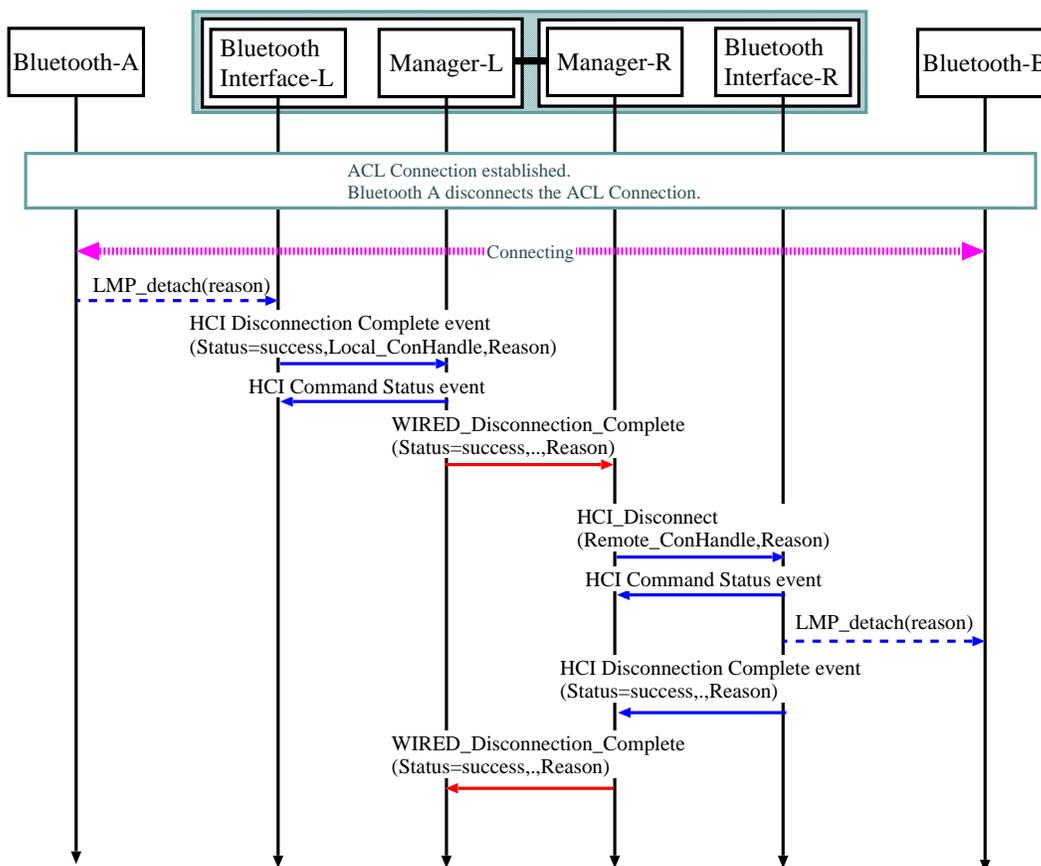


図 6.10: ACL 切断

6.4 ACL 接続確立後のアクティビティ

6.4.1 認証の要求

図 6.11 に ACL 確立後の認証の要求に対するメッセージ・シーケンス・チャートを示す。ACL 確立後に認証の要求を受けた場合、ローカル側の Manager へは HCI Link Key Request イベントが発生する。このイベントに対して、リモート側 Manager は認証を行っていない場合、HCI Authentication Request コマンドを実行することで、リモート側における ACL 確立後の認証の実現する。

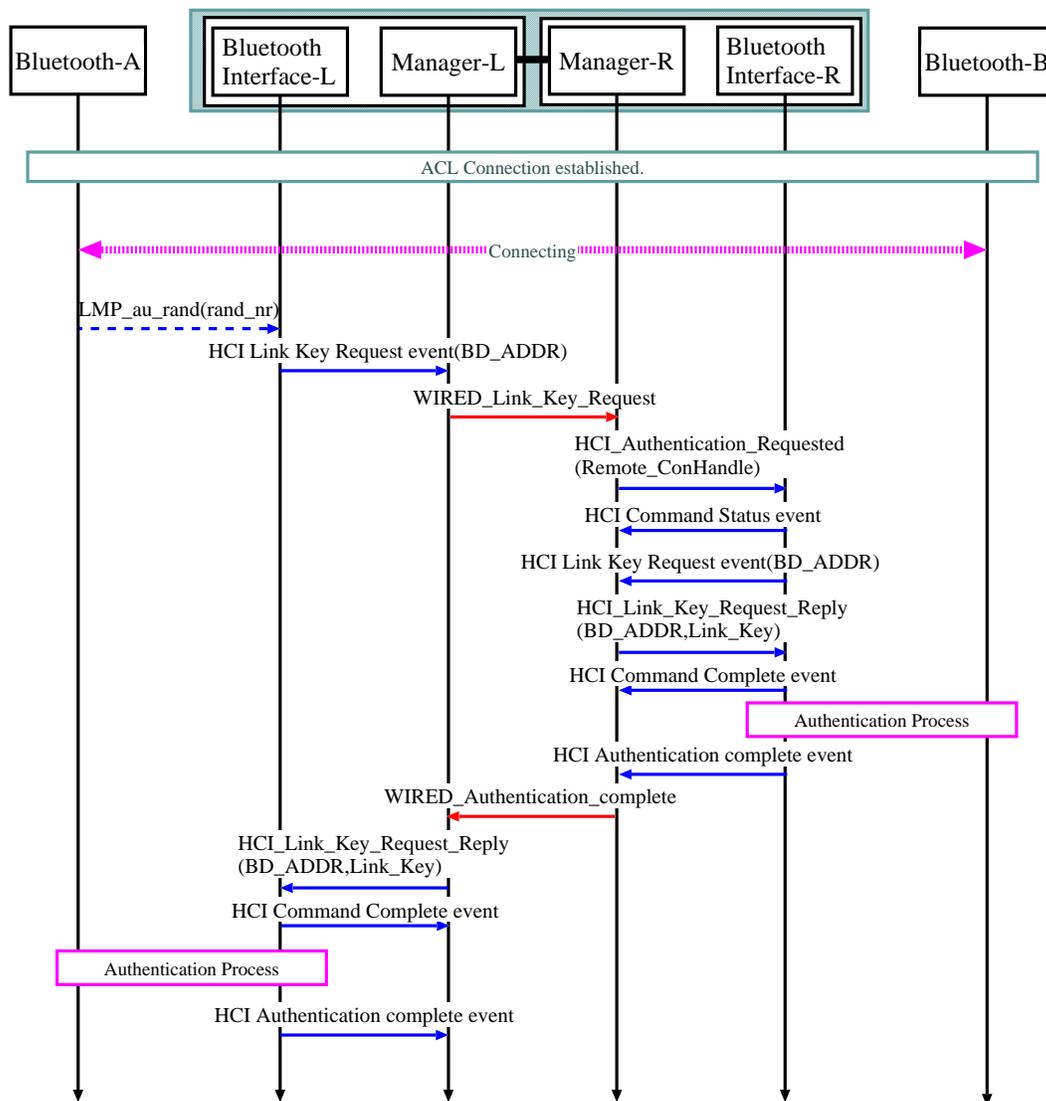


図 6.11: 認証の要求

6.4.2 接続の暗号化の設定

図 6.12 に ACL 確立後の暗号化の要求に対するメッセージ・シーケンス・チャートを示す。ACL 確立後に暗号化の要求を受けた場合、ローカル側の Manager へは HCI Encryption Change イベントが発生する。このイベントに対して、リモート側 Manager が、HCI_Set_Connection_Encryption コマンドを実行することで、リモート側における ACL 確立後の暗号化を実現する。

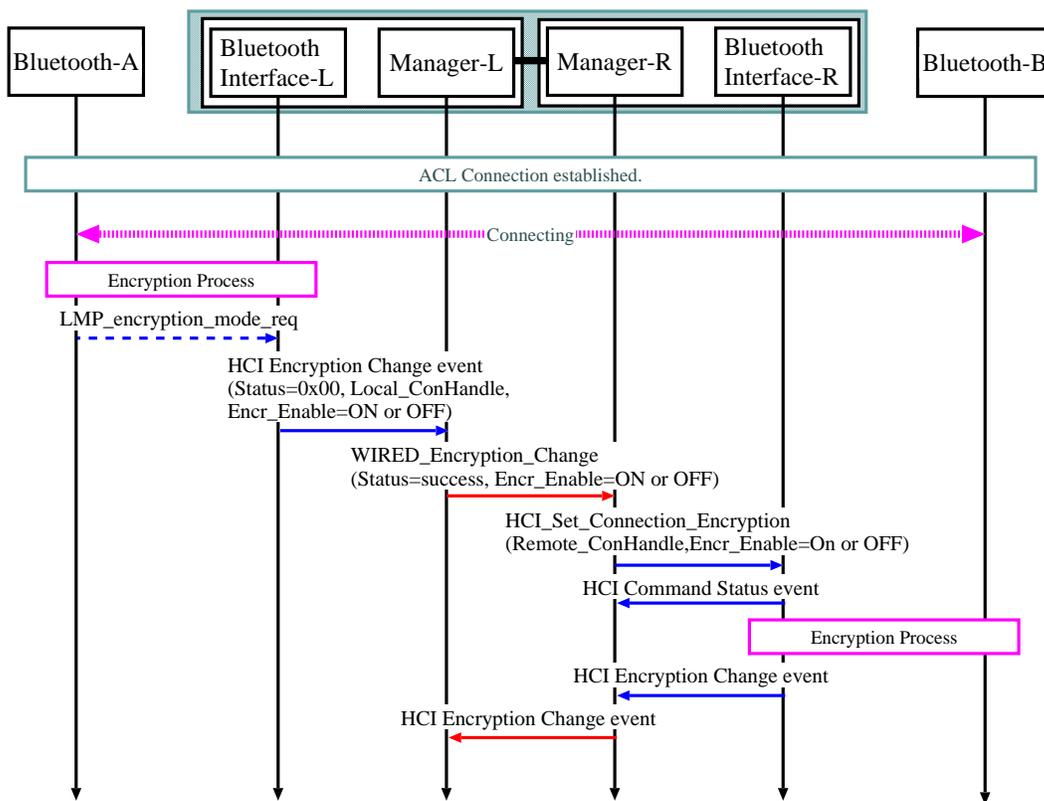


図 6.12: 接続の暗号化の設定

6.4.3 接続リンク・キーの変更

図 6.12 に ACL 確立後の接続リンク・キーの変更要求に対するメッセージ・シーケンス・チャートを示す。ACL 確立後に接続リンク・キーの変更要求を受けた場合、ローカル側の Manager へは HCI Link Key Request イベントが発生する。このイベントに対して、リモート側 Manager はすでに認証を行っている場合は接続リンク・キーの変更要求であると判断し、HCI_Change_Connection_Link_Key コマンドを実行することで、リモート側における ACL 確立後の接続リンク・キーの変更を実現する。

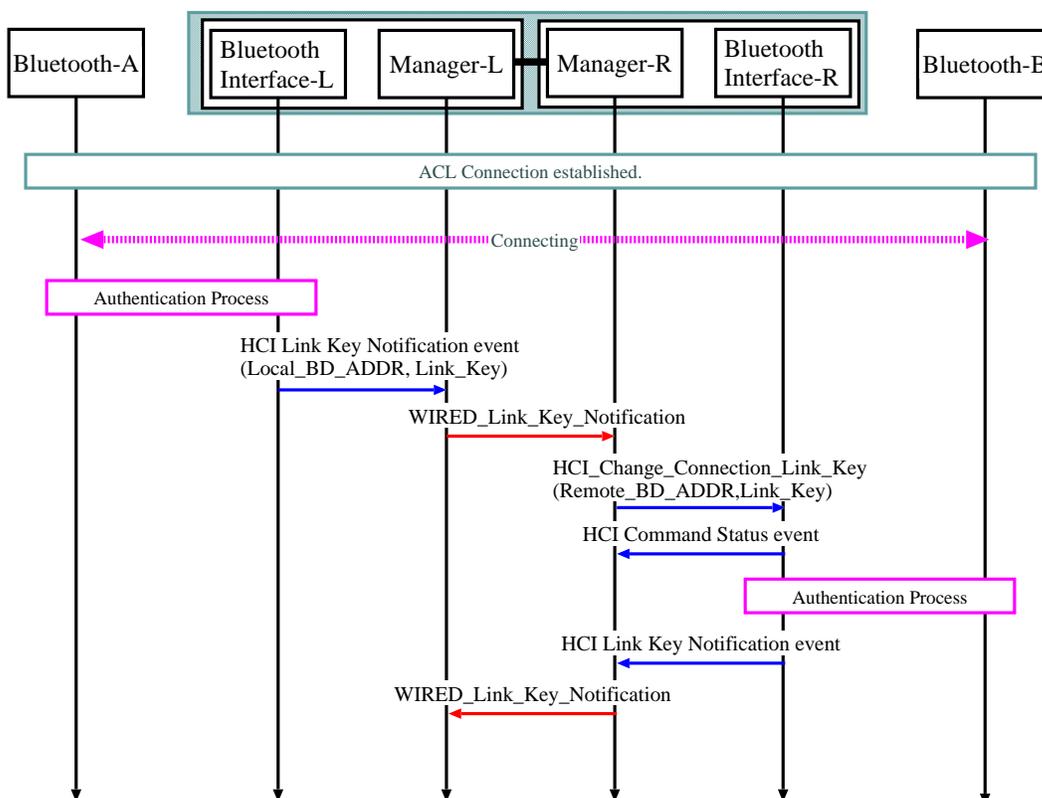


図 6.13: 接続リンク・キーの変更

6.4.4 マスター・リンク・キー

図 6.14 に ACL 確立後のマスター・リンク・キーの変更要求に対するメッセージ・シーケンス・チャートを示す。ACL 確立後にマスター・リンク・キーの変更要求を受けた場合、ローカル側の Manager へは HCI Master Link Key Complete イベントが発生する。このイベントに対して、リモート側 Manager は、HCI_Master_Link_Key コマンドを実行することで、リモート側における ACL 確立後のマスター・リンク・キーの変更を実現する。

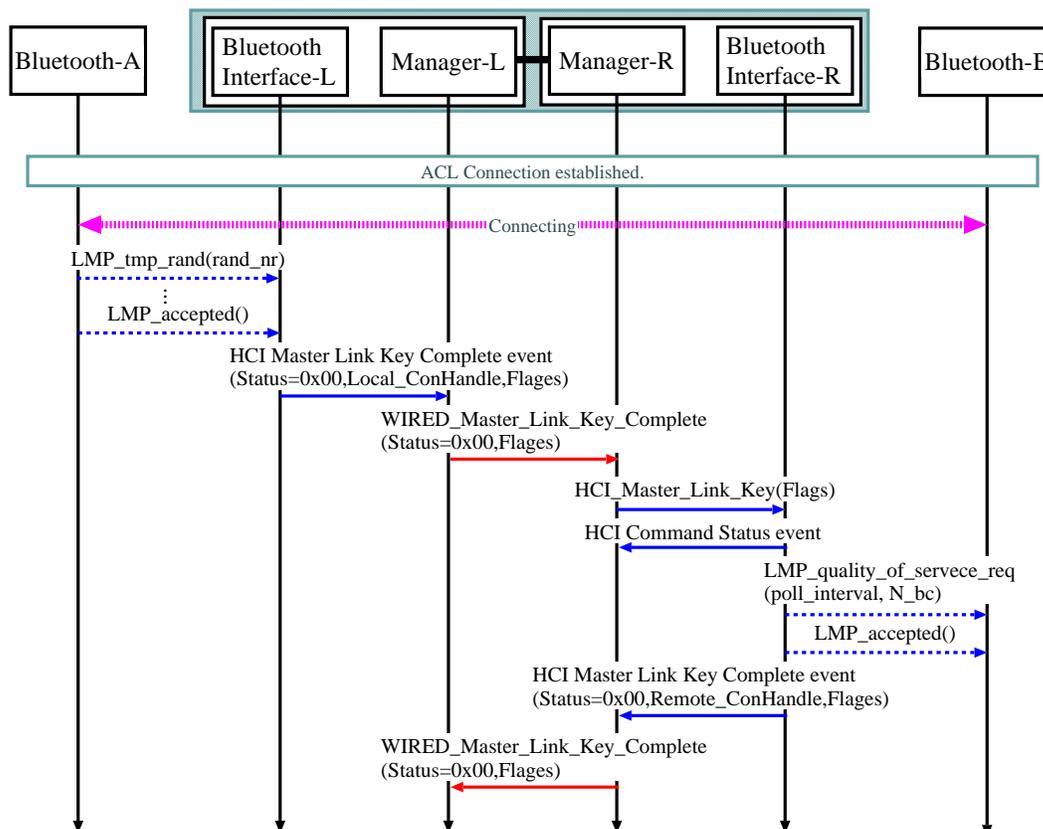


図 6.14: マスター・リンク・キー

6.4.5 QoS のセットアップ

図 6.15 に ACL 確立後の QoS セットアップの変更要求に対するメッセージ・シーケンス・チャートを示す。ACL 確立後に QoS セットアップの要求を受けた場合、ローカル側の Manager へは HCI QoS Setup Complete イベントが発生する。このイベントに対して、リモート側 Manager は、HCI_QoS_Setup コマンドを実行することで、リモート側における ACL 確立後の QoS セットアップの要求を実現する。QoS Setup Complete イベントから得られるパラメータは、ローカル側の提案システムを用いた場合、QoS Setup Complete イベントから得られるパラメータは、LM 層でネゴシエーションした結果の値であり、リモート側での HCI_QoS_Setup コマンドのパラメータへは得られたパラメータを用いる必要があるがリモート側の LM 層のネゴシエーション次第で、完全に一致させることができるとは限らない。

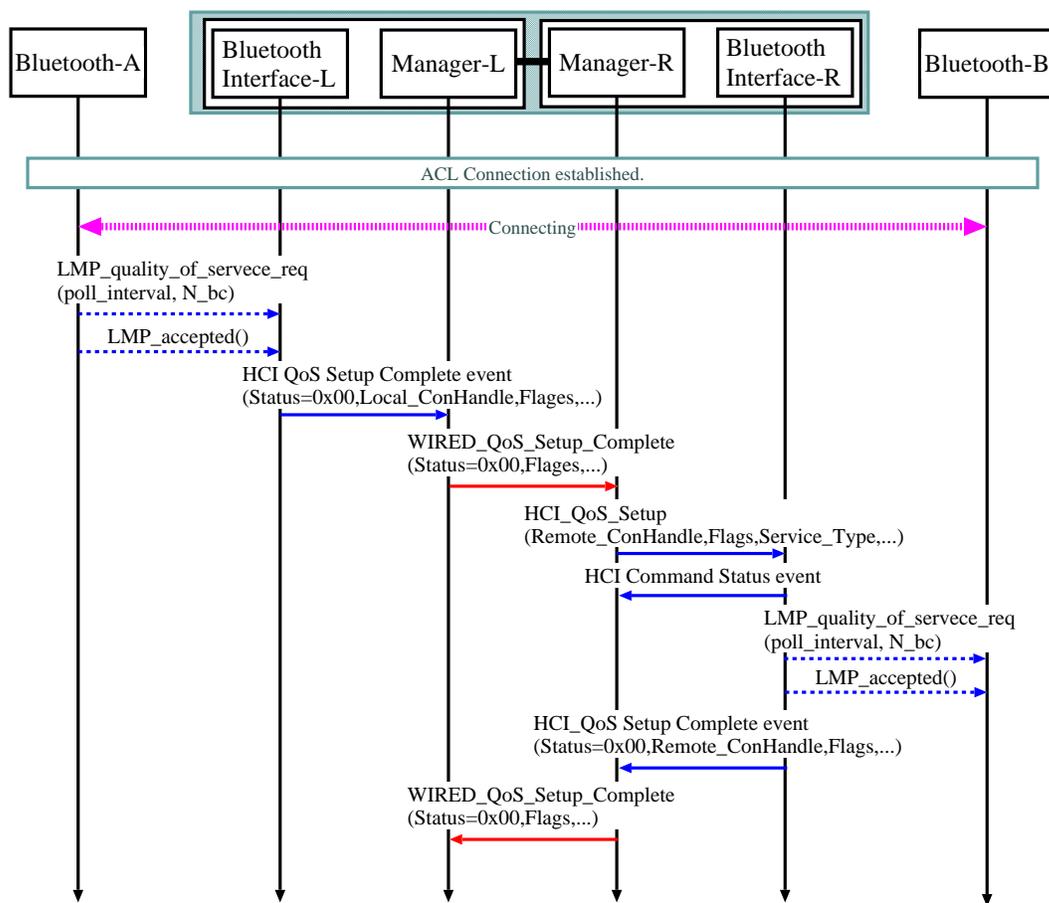


図 6.15: QoS のセットアップ

6.4.6 役割の切り替え

図 6.16 に ACL 確立後におけるマスター・スレーブの役割の変更要求に対するメッセージ・シーケンス・チャートを示す。ACL 確立後に役割の切り替え要求を受けた場合、ローカル側の Manager へは HCI Role Change イベントが発生する。このイベントに対して、リモート側 Manager は、HCI_Role_Change コマンドを実行することで、リモート側における ACL 確立後の QoS セットアップの要求を実現する。

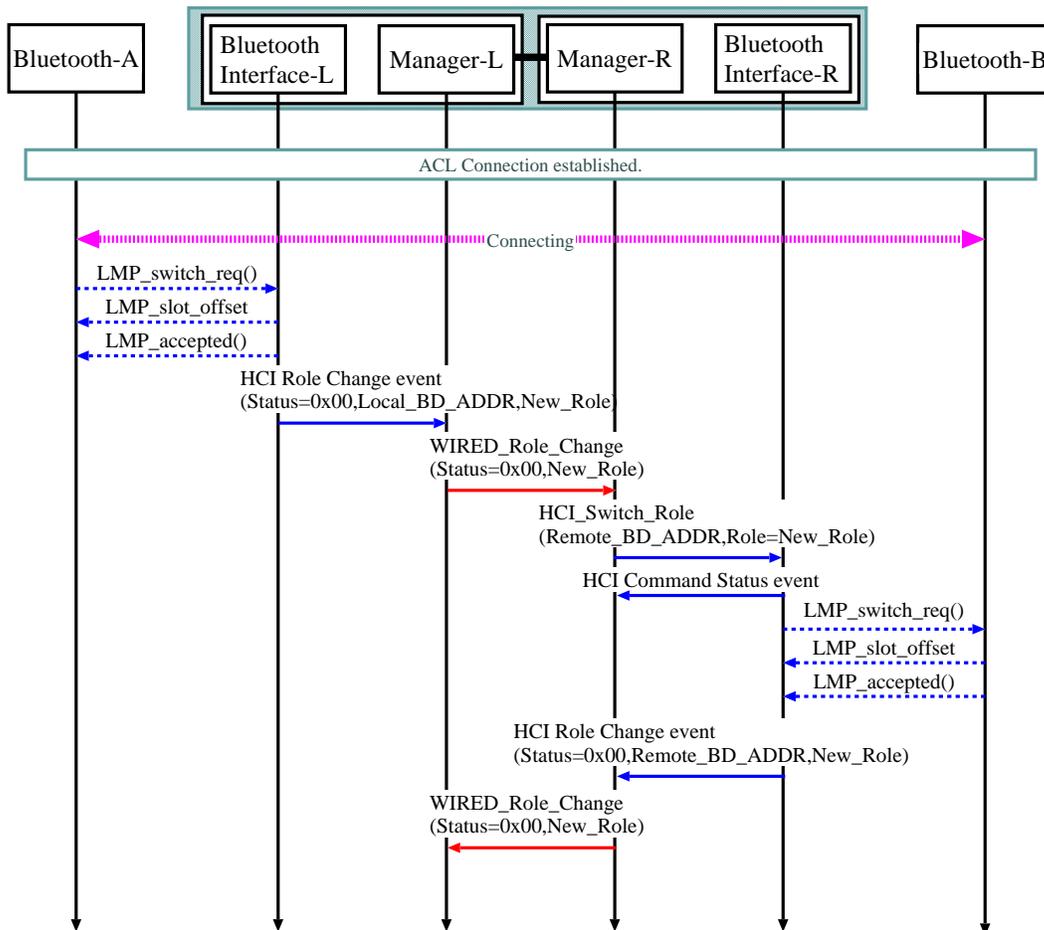


図 6.16: 役割の切り替え

6.5 SCO 接続の確立と切断

6.5.1 SCO 接続セットアップ

図 6.17 に SCO 確立時のメッセージ・シーケンス・チャートを示す。SCO 接続セットアップのためのメッセージ・シーケンスは、ACL とほぼ同様である。

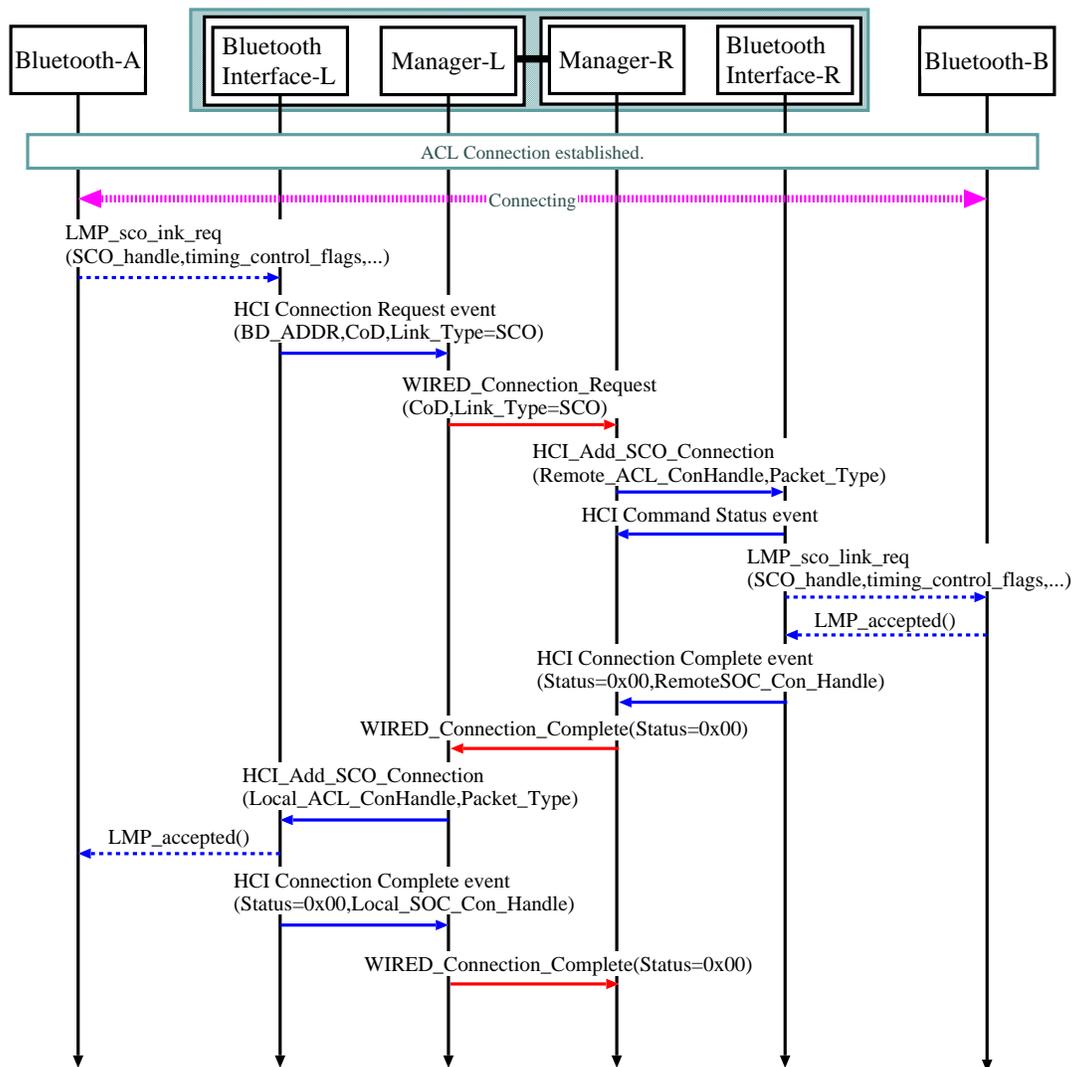


図 6.17: SCO 接続セットアップ

6.5.2 SCO 切断

図 6.18 に SCO 確立時のメッセージ・シーケンス・チャートを示す。SCO 切断のためのメッセージ・シーケンスは、ACL とほぼ同様である。

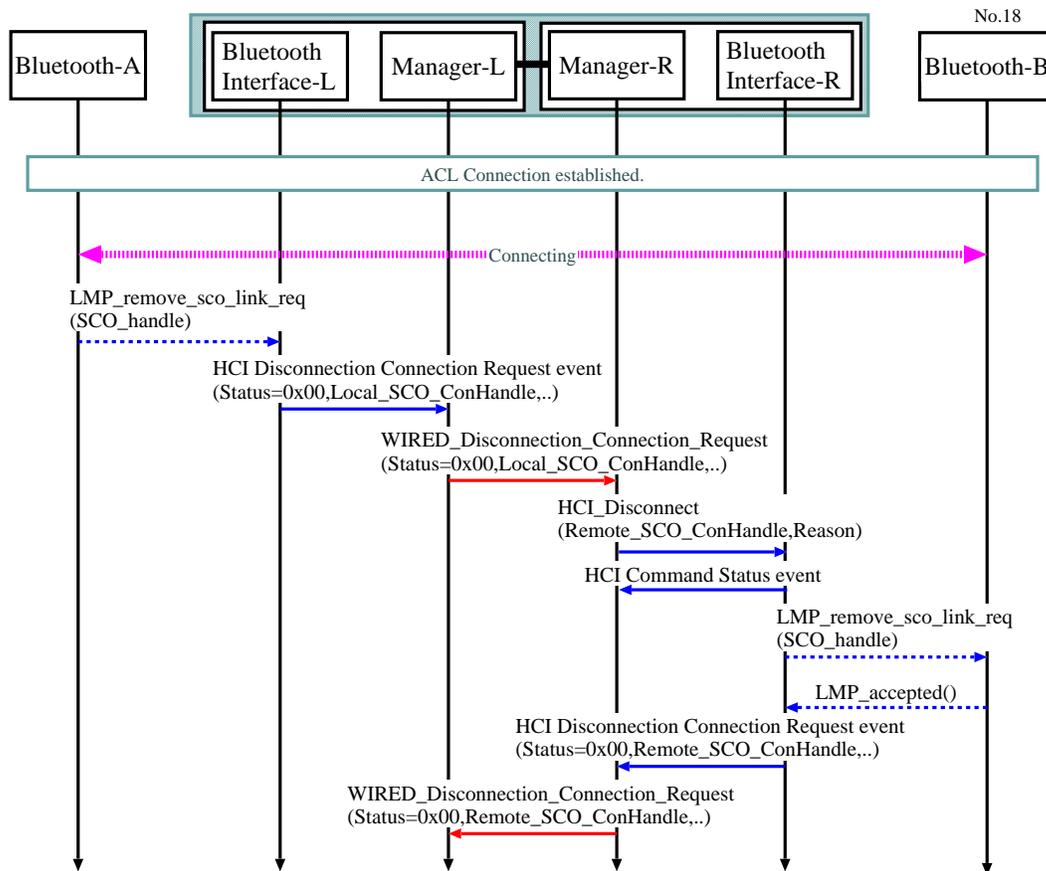


図 6.18: SCO 切断

6.6 特別なモード

6.6.1 Sniff モード

図 6.19, 6.20 に Sniff モードへのモード変更時, および終了時のメッセージ・シーケンス・チャートを示す. ACL 確立後に Sniff モードへの変更要求を受けた場合, ローカル側の Manager へは HCI Mode Change イベントが発生する. このイベントのパラメータ Current_Mode が Sniff モードであった場合, リモート側 Manager は, HCLSniff_Mode コマンドを実行することで, リモート側における ACL 確立後の QoS セットアップの要求を実現する. 提案システムを用いた場合, Mode Change イベントから得られるパラメータは, LM 層でネゴシエーションした結果の値であり, リモート側での HCLSniff_Mode コマンドのパラメータへは得られたパラメータを用いる必要があるがリモート側の LM 層のネゴシエーション次第で, 完全に一致させることができるとは限らない. また, 片側で Sniff モードが終了した際には HCI Mode CHange イベントによって Sniff モードから通常のモードへの変更が通知され, もう一方で HCLExit_Sniff_Mode を実行し, Sniff モードの終了を行う. また, これらのモードの変更に対応するために, 提案システムは常に現在のモードを把握する必要がある.

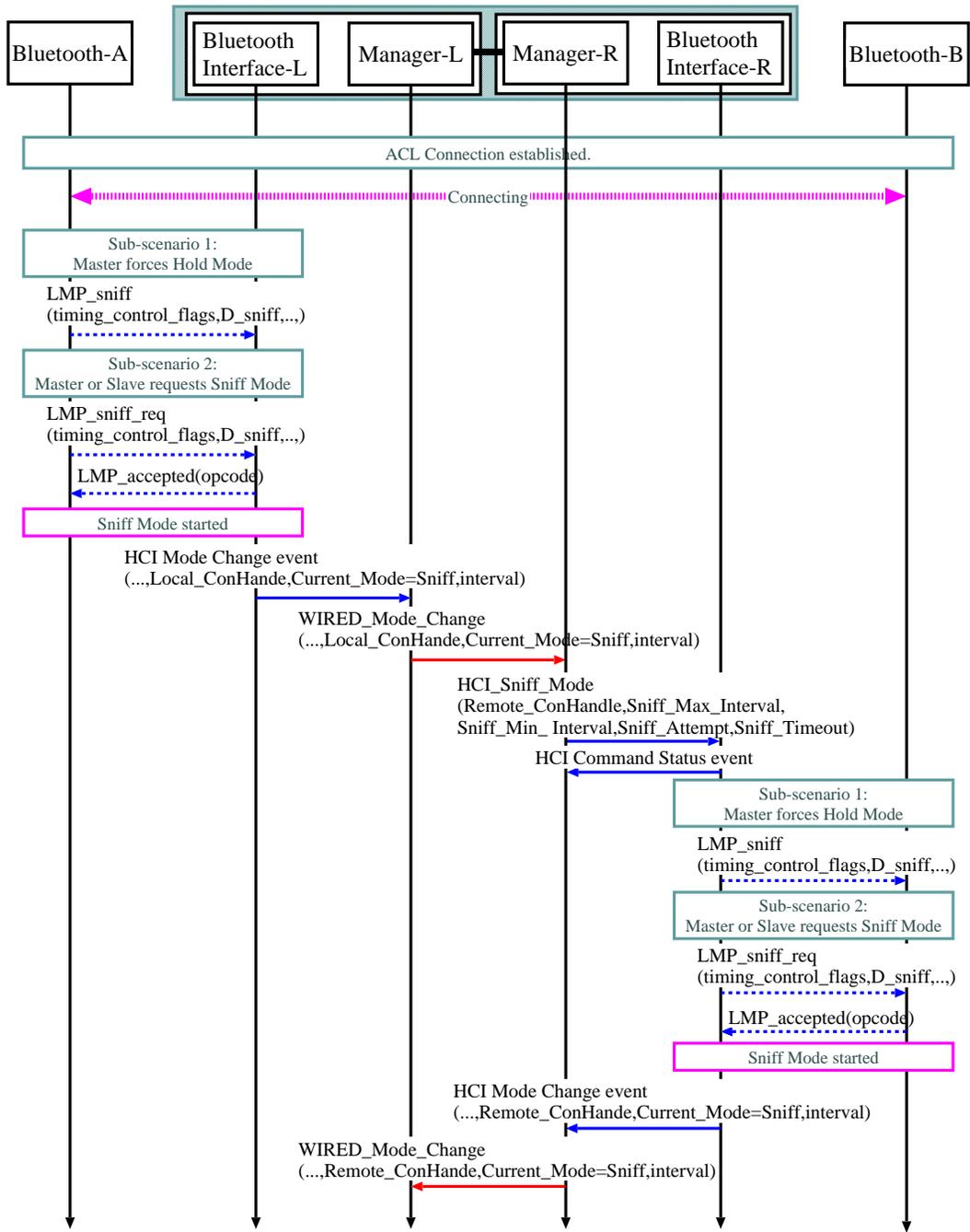


図 6.19: Sniff モード

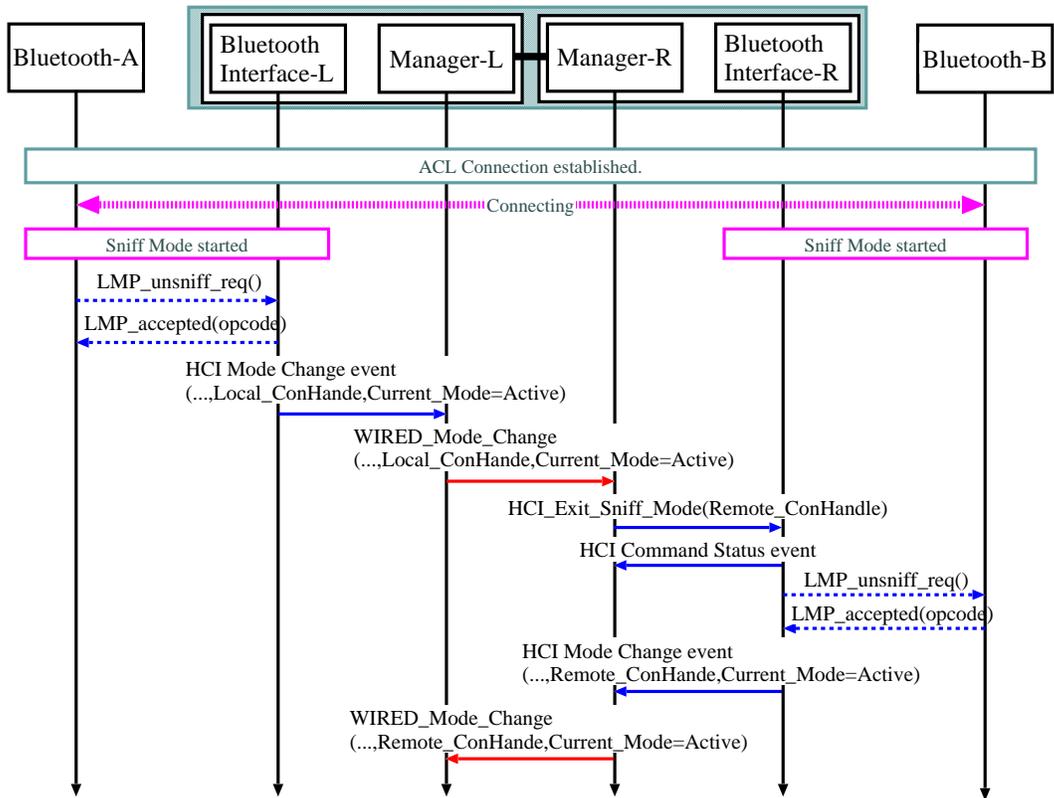


図 6.20: Sniff モード 終了

6.6.2 Hold モード

図 6.21 に Hold モードへのモード変更時のメッセージ・シーケンス・チャートを示す。このイベントのパラメータ `Current_Mode` が Hold モードであった場合、リモート側 Manager は、`HCLHold_Mode` コマンドを実行することで、リモート側における ACL 確立後の QoS セットアップの要求を実現する。提案システムを用いた場合、Mode Change イベントから得られるパラメータは、LM 層でネゴシエーションした結果の値であり、リモート側での `HCLHold_Mode` コマンドのパラメータへは得られたパラメータを用いる必要があるがリモート側の LM 層のネゴシエーション次第で、完全に一致させることができるとは限らない。また、これらのモードの変更に対応するために、提案システムは常に現在のモードを把握する必要がある。

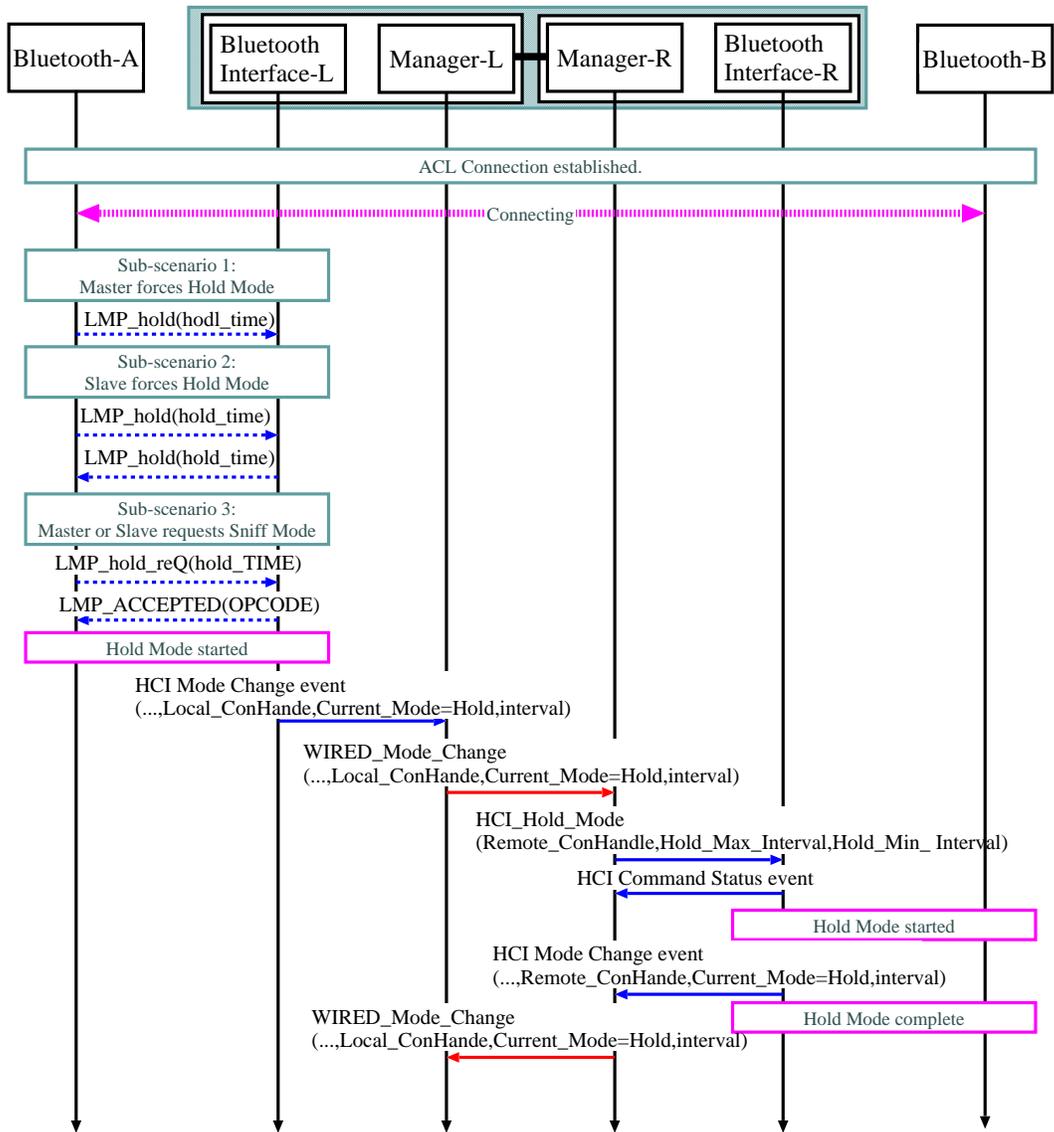


図 6.21: Hold モード

6.6.3 Park モード

図 6.22, 6.22 に Sniff モードへのモード変更時, および終了時のメッセージ・シーケンス・チャートを示す.

このイベントのパラメータ `Current_Mode` が Park モードであった場合, リモート側 Manager は, `HCI_Park_Mode` コマンドを実行することで, リモート側における ACL 確立後の QoS セットアップの要求を実現する. 提案システムを用いた場合, Mode Change イベントから得られるパラメータは, LM 層でネゴシエーションした結果の値であり, リモート側での `HCI_Park_Mode` コマンドのパラメータへは得られたパラメータを用いる必要があるがリモート側の LM 層のネゴシエーション次第で, 完全に一致させることができるとは限らない. また, 片側で Park モードが終了した際には `HCI_Mode_Chang` イベントによって Park モードから通常モードへの変更が通知され, もう一方で `HCI_Exit_Park_Mode` を実行し, Park モードの終了を行う. これらのモードの変更に対応するために, 提案システムは常に現在のモードを把握する必要がある.

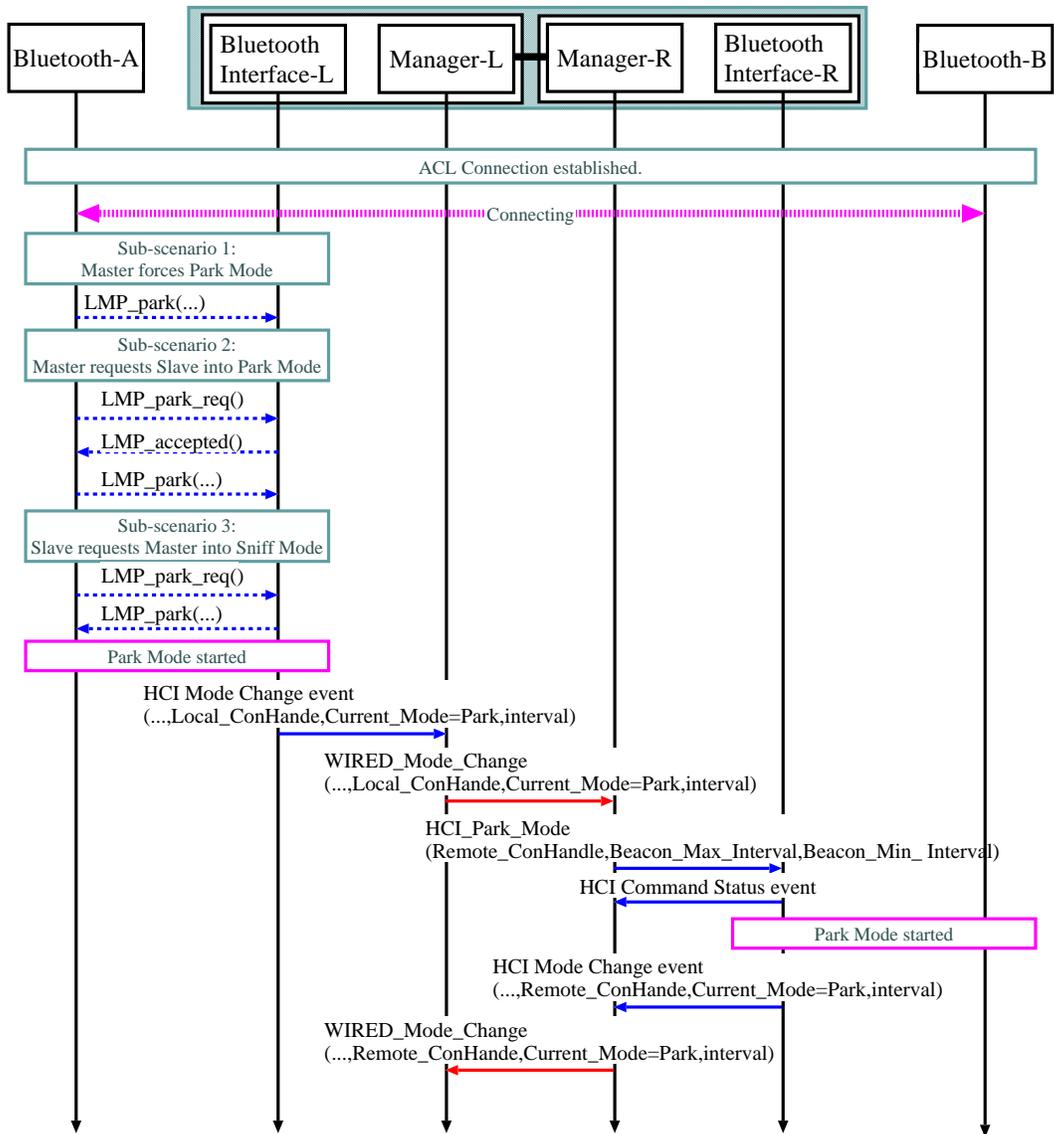


図 6.22: Park モード

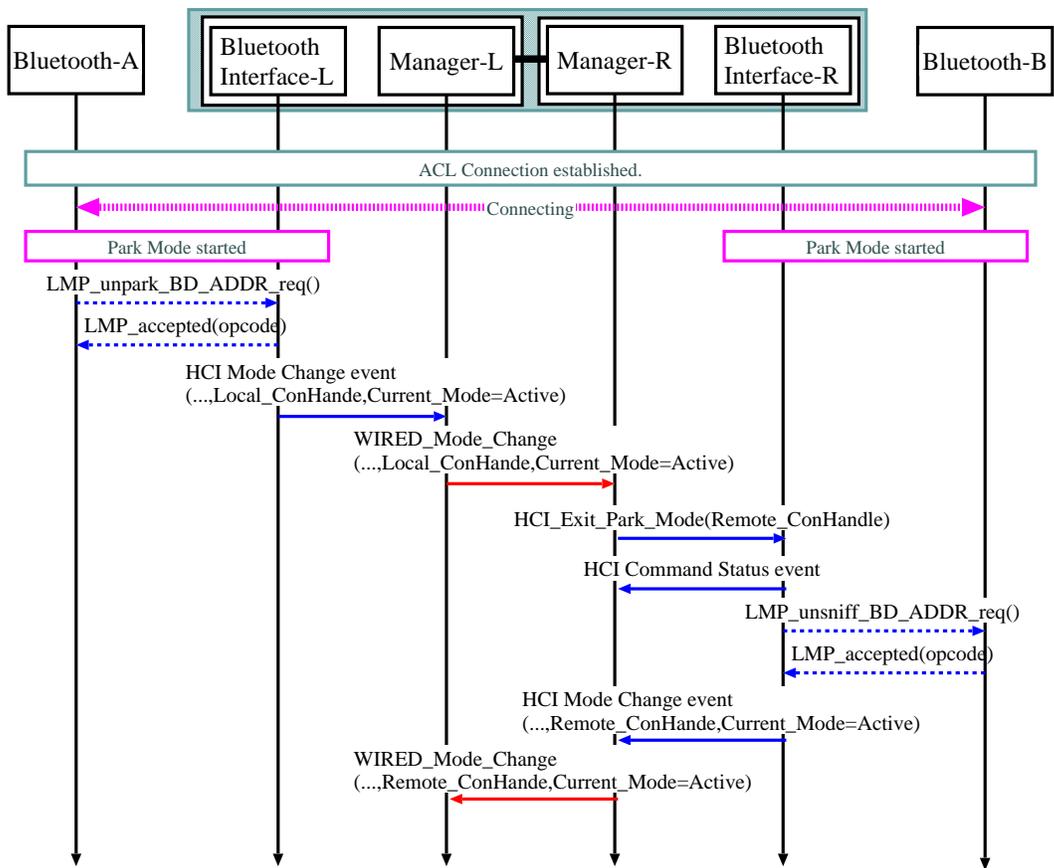


図 6.23: Park モードの終了

第7章 提案システムの実装

提案する有線接続システムの実装を行い、その評価として、提案システムによる既存の Bluetooth 機器に対する接続検証実験、および接続時の伝送遅延、スループット、通信リンク確立までの時間に関して評価を行った。

7.1 実装

提案する有線接続システムの実装および評価ツールの実装を行った。実装した提案システムを図 7.1 に示す。提案システムは、USB インターフェースおよび Ethernet インターフェースをもつラップトップ PC 上に Bluetooth USB モジュールを接続し Linux 上に実装した。また、提案システムを評価するための Bluetooth 機器として評価ツールを同様に



図 7.1: 実装した提案システム

ラップトップ PC 上に実装し、これらの機器を用いて提案システムの評価を行った。図 7.2 に実装した提案システムおよび評価用の Bluetooth 機器の実装環境を示す。提案システムの Manager は、Linux Kernel 2.4-24 上にソフトウェアにより実装し、また Bluetooth インターフェースとして提案システムに 3Com Bluetooth USB Adapter、評価用のラップトッ

PC に HAGIWARA SYS-COM Bluetooth USB Stick を用いた。それぞれの Bluetooth モジュールのドライバは、Linux BlueZ [8] を用いた。

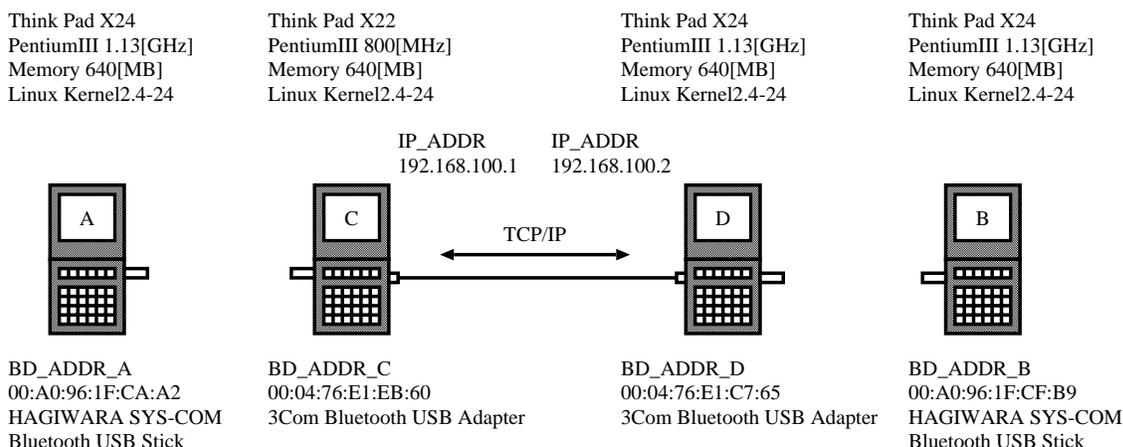


図 7.2: 実装した提案システムの実装環境

実装した提案システムは、ローカル側、リモート側にそれぞれ1つの Bluetooth インターフェースを持ち、Bluetooth 機器をローカル側で1台、リモート側で1台を接続可能なものを実装した。実装した提案システムのソフトウェア構成を図 7.3 に示す。

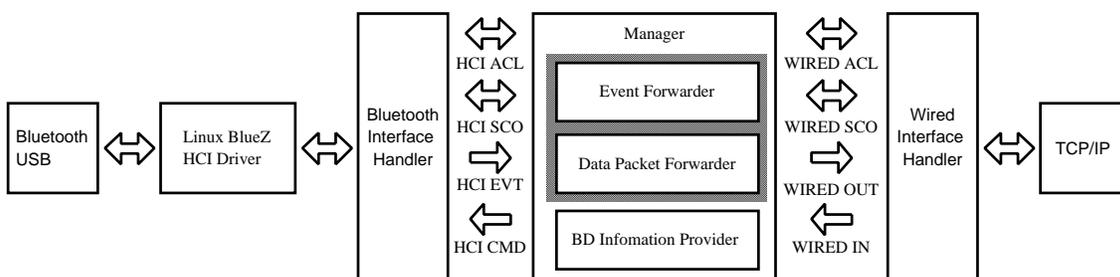


図 7.3: 実装した提案システムのソフトウェア構成

各コンポーネントの概要を以下に示す。

Linux BlueZ HCI Driver

Linux BlueZ の HCI ドライバー。Bluetooth モジュールの HCI に対する socket インターフェースを提供する。

Bluetooth Interface Handler

Linux BlueZ HCI Driver からの HCI イベントパケット、HCI データパケットを Manager へ提供する、また、Manager からの HCI コマンドパケットの実行を行う。

Wired Interface Handler

有線伝送路上のメッセージパケット、データパケットに関する処理を行う。Manager からのメッセージ、データパケットを接続先の提案システムが識別するためのヘッダを加えて送信する。また、有線伝送路から届いたメッセージ、データパケットを受信し、パケットの識別を行い、Manager に提供する。

Event Forwarder

Manager のコンポーネントの一部。Bluetooth Interface Handler および Wired Interface Handler からの HCI イベントパケット、メッセージパケットを受信し、対応する HCI コマンドの実行、リモート側へのメッセージパケットの送信を行うことで、HCI イベントの転送を行う。

Data Packet Forwarder

Manager のコンポーネントの一部。Bluetooth Interface Handler および Wired Interface Handler からの HCI ACL データパケット、HCI SCO データパケット、有線伝送路からの ACL データパケット、SCO データパケットの転送処理を行う。有線伝送路からのデータパケットに対して、対応する Bluetooth 機器の通信リンクのコネクションハンドルを指定し、データパケットの送信を行う。

BD Information Provider

提案システムの周辺に存在する Bluetooth 機器の情報の収集および取得した情報の有線伝送路上への送信処理を行う。また、有線伝送路から受信した情報を自分の持つ Bluetooth インターフェースに反映することで、ローカル側の Bluetooth 機器に対してリモート側の Bluetooth 機器の情報を提供する。

7.2 既存の Bluetooth 機器に対する接続検証

実装した提案システムを用いて、既存の Bluetooth 製品に対する提案システムの接続検証を行った。検証を行った結果、提案する有線接続システムによる既存の Bluetooth 製品の透過接続の実現を確認できた。検証を行った Bluetooth 製品は次の製品である。

- SONY Syber-shot DSC-FX77 (Basic Image プロファイル)
- EPSON Color Printer PM-860PT (Basic Image プロファイル)
- HAGIWARA Bluetooth Suite (Object Exchange プロファイル)
- HAGIWARA Bluetooth Suite (LAN アクセスプロファイル)
- Palm Bluetooth アプリケーション (Object Exchange プロファイル)

7.3 実装した提案システムの評価

7.3.1 コネクション確立までの遅延

提案システムを用いた Bluetooth 機器間の接続を行った際に L2CAP 層におけるコネクション確立までの時間を測定した。この測定の目的は、提案システムを用いることで発生する遅延によって接続対象である Bluetooth 機器がコネクションタイムアウトを発生させないことの確認、および、どの程度の遅延が発生するのかを実装した提案システムを用いて測定することである。

測定は、比較のため表 7.1 に示すように、Bluetooth 機器間を 1m の距離で直接接続した場合と、提案システムを用いて Bluetooth 機器と Bluetooth インターフェース間を 1m の距離、有線伝送路を 5m の距離で接続した場合の 2 通りの測定を行った。図 7.4 に実装した提案システムと Bluetooth 機器の配置図を示す。図 7.4 中の A, B, C, D は図 7.2 の各機器に対応している。それぞれの場合において、Bluetooth A から Bluetooth B に対して L2CAP 層での接続要求を行い、コネクション確立までにかかった時間を測定した。

接続方法	接続距離
直接	1m
提案システムによる有線接続	Bluetooth インターフェース間 1m Bluetooth 有線伝送路 5m

表 7.1: 直接接続した場合との評価環境

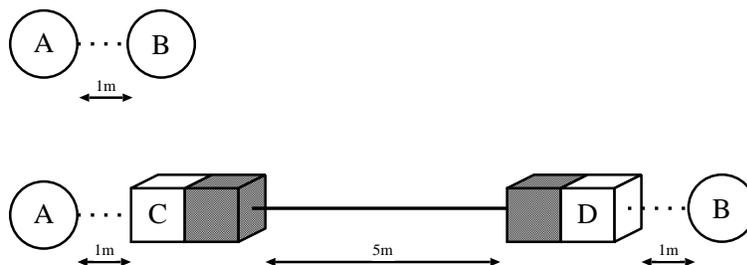


図 7.4: 提案システムと Bluetooth 機器の配置図

また、測定は、直接接続と提案システムを用いた場合のそれぞれの場合において Bluetooth A と Bluetooth B のセキュリティモードを以下の設定で行った。

- Bluetooth A:セキュリティなし, Bluetooth B:セキュリティなし
- Bluetooth A:認証・暗号化要求, Bluetooth B:認証・暗号化要求
- Bluetooth A:セキュリティなし, Bluetooth B:認証・暗号化要求

	A	B	直接接続 (1m)	提案システム	遅延
セキュリティ	なし	なし	1.88 sec	3.37 sec	1.49 sec
	暗号化 認証	暗号化 認証	2.08 sec	4.28 sec	2.20 sec
	なし	暗号化 認証	2.04 sec	4.19 sec	2.15 sec
	暗号化 認証	なし	2.04 sec	6.11 sec	4.07 sec

表 7.2: L2CAP コネクション確立までの平均時間

	A	B	直接接続 (1m)	提案システム	遅延
セキュリティ	なし	なし	2.39 sec	3.42 sec	1.03 sec
	暗号化 認証	暗号化 認証	2.43 sec	5.46 sec	3.03 sec
	なし	暗号化 認証	2.58 sec	4.43 sec	1.85 sec
	暗号化 認証	なし	2.09 sec	6.17 sec	4.08 sec

表 7.3: L2CAP コネクション確立までの測定された最大時間

- Bluetooth A:認証・暗号化要求, Bluetooth B:セキュリティなし

これは、前章においてメッセージ・シーケンス・チャートで示した通り提案システムは、ACL リンクのコネクション確立時にセキュリティモードの同期をとるために上記の条件でそれぞれ異なるメッセージのやりとりを行うためである。直接接続と提案システムを用いた場合において L2CAP 層でのコネクション要求からコネクション確立までにかかった時間を 30 回測定した結果の平均時間を表 7.2、測定結果中の最大時間を表 7.3 にまとめる。これらのコネクション確立時間には、L2CAP 層よりも下位層のリンクが確立されていない状態からの baseband 層および LM 層における ACL リンク確立処理にかかる時間も含まれている。

平均値および最大値から提案システムを用いた場合、L2CAP 層におけるコネクションを確立するためには、直接接続した場合と比べ、約 1 秒から 4 秒程度の遅延が発生することがわかった。このとき、提案システムにより遅延が発生した場合においても、コネクションの確立に失敗することはなかった。また、提案システムを用いた場合、接続される機器のセキュリティー・モードによって接続にかかる時間が異なっていることが確認できる。とくに接続要求側の Bluetooth A が認証・暗号化を要求し、セキュリティー・モードを要求しない Bluetooth B に提案システムを介して接続する際に最も大きくなっている。これ

は、この組み合わせの場合にリモート側において1度確立した ACL リンクを切断し、再接続する(図 6.9) ために遅延が大きくなるためである。すべての接続の試行に対してコネクションの確立に成功したことで、提案システムによる遅延が接続する Bluetooth 機器にとって十分に少ないことが言える。

さらに、L2CAP 層から考察すると、コネクション要求時のタイマーである RTX(Response Timeout Expired) タイマの値が最小 1 秒、最大 60 秒であり、このタイマーが 7 秒以上ならば接続可能であるという結果は、十分に速い速度でコネクション要求に対するレスポンスを返すことが出来ているといえる。これは、RTX は、通常接続相手が PIN 入力を要求する時間を考慮し 30 秒以上の設定となっている場合が多いためである。

7.3.2 データパケットの伝送遅延

提案システムを用いて Bluetooth 機器間の接続を行った場合におけるデータの伝送遅延特性を調べるために、L2CAP 層における Echo Request, Echo Response を利用した Ping よるラウンド・トリップ・タイムの測定を行った。図 7.5 に図 7.4 と同様に 1m の距離で直接接続した場合と提案システムを用いて接続した場合における Bluetooth A から Bluetooth B へのラウンド・トリップ・タイムを示す。図 7.5 から提案システムを用いて接続した場合と直接接続した場合では、約 10msec から 30msec の差がみられ、片方向で約 5msec から 15msec の遅延が発生することがわかる。L2CAP 層では、データパケットに対するタイマーが存在しないため、提案システムを用いることで発生する遅延による問題は、L2CAP 層より上位のプロファイルやアプリケーションのデータパケットに対するタイムアウト値次第である。例として、L2CAP プロトコル上にシリアル・ポートをエミュレーションする RFCOMM プロファイルを挙げると、このプロファイルのもつ確認応答タイマーの値は、10 から 60 秒(推奨値 20 秒)である。この値から、提案システムによる約 5msec から 15msec の遅延は接続される Bluetooth 機器にとって十分に高速であると考えられる。これらのことから提案システムによる有線接続手法を用いることで、かなり距離の離れた Bluetooth 機器間も接続することが可能であるといえる。

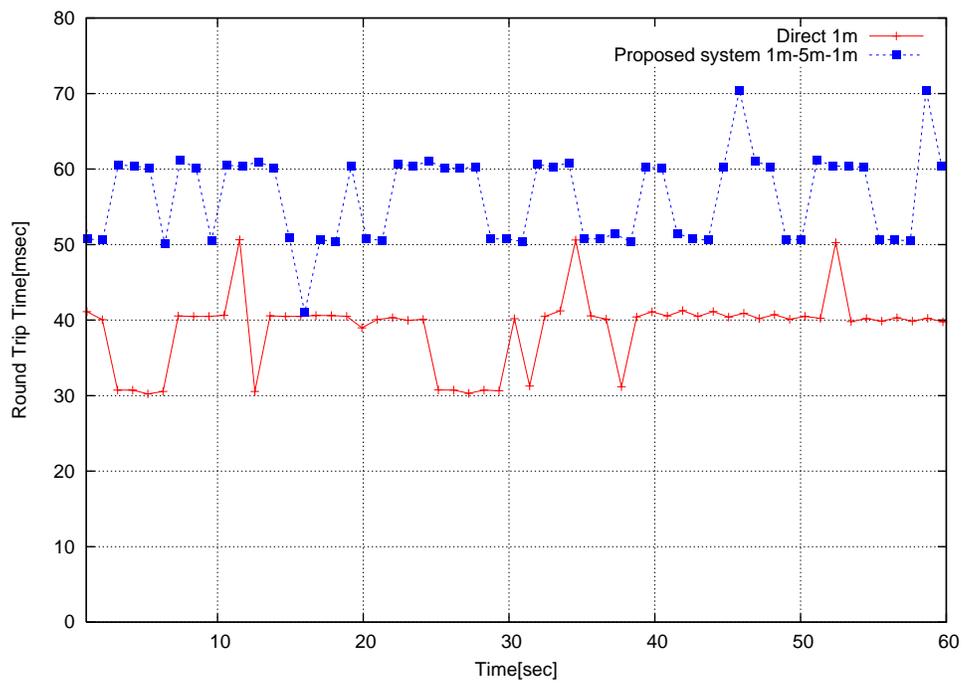


図 7.5: 提案システムの遅延

7.3.3 スループット

提案システムを用いて Bluetooth 機器の接続を行った場合における Bluetooth 機器間のスループットの測定を行った。スループットは、672 バイトの L2CAP パケットを図 7.4 と同様に 1m の距離で直接接続した場合と提案システムを用いて接続した場合において Bluetooth A から Bluetooth B へ連続で送信し、1 秒間に Bluetooth B が受信したデータ数を DM1 から DH5 までの各パケットタイプを用いてそれぞれの場合において測定した。また、測定の際、提案システムを用いて接続した場合におけるローカル側とリモート側の ACL リンクの パケットタイプは、予め同一のパケットタイプを利用する設定を行い、使用されるパケットタイプを一致させた。

DM1 パケット利用時のスループット

図 7.6 に、1m の距離で直接接続した場合と提案システムを用いて接続した場合における Bluetooth A から Bluetooth B への DM1 パケット利用時のスループットを示す。DM1 パケットを用いた ACL リンクの順方向への最大スループットは 108.8Kbps であるので、1m の近距離で接続した直接接続ではほぼ最大スループットを得ることができている。これに対して、提案システムを用いて接続した場合は、最大スループットの約 70% 程度のスループットしか得ることができていない。このスループットの低下は、Bluetooth インターフェースからの HCI データパケットのサイズとパケット数の違いから発生すると考えられる。表 7.4 は、提案システムを用いて接続した場合において、各パケットタイプを用いた際に、ローカル側の Bluetooth インターフェースから Manager へ渡される HCI ACL データパケットの平均データ長と 1 秒間に処理されるパケット数である。表 7.4 から、DM1 のパケットタイプを用いた場合、Bluetooth インターフェースから渡される ACL パケットの平均データ長が 16byte と小さいにもかかわらず、1 秒間に処理しなくてはならないパケット数は DM3 などのパケットタイプを用いた場合に比べ、760~780 packet と非常に多いことがわかる。これは、baseband 層における DM1 パケットタイプのペイロードが 18byte(1byte のペイロード・ヘッダを含む) と非常に小さいことから、HCI から渡される HCI データパケットが小さなまま渡されるためである。提案システムのリモート側の Bluetooth インターフェースである Bluetooth モジュールのバッファ数には限りがあり、データが接続相手に送信されるまで決まった数の HCI データパケットしか入力できない。このため、当然たくさんのデータを送信したい場合は、大きなサイズの HCI データパケットを複数入力することが望ましい。しかし、DM1 のパケットタイプを用いた場合は小さいパケットを多数処理しなくてはならないことから、提案システムに接続された Bluetooth モジュールの性能を發揮できない。スループットの低下はこのために発生しているものと考えられる。提案システム内におけるデータパケットの合成など、処理の最適化を行うことで、最終的には後述する DM3, DM5 のパケットタイプを利用した場合と同様に直接接続と同等のスループットを得ることができると考えられる。

	ACL パケットの平均データ長	パケット数/秒
DM1 パケット	16 byte	760 ~ 780 packet
DH1 パケット	26 byte	760 ~ 780 packet
DM3 パケット	112 byte	370 ~ 380 packet
DH3 パケット	96 byte	650 ~ 670 packet
DM5 パケット	112 byte	370 ~ 380 packet
DH5 パケット	112 byte	600 ~ 620 packet

表 7.4: ACL パケットの平均データ長と 1 秒間に処理されるパケット数

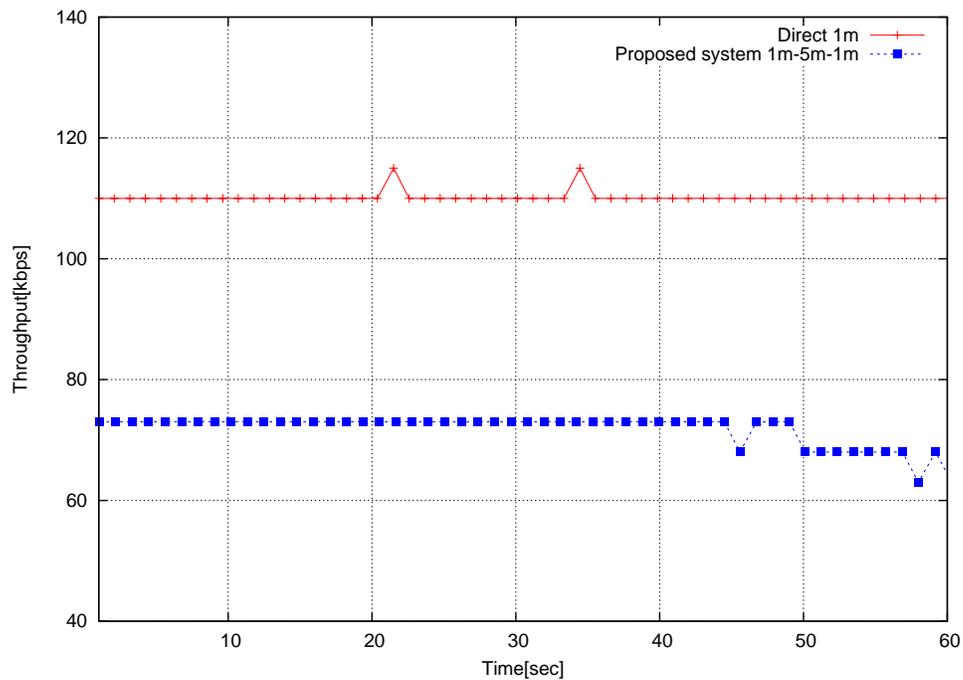


図 7.6: 提案システムのスループット (DM1 パケット使用時)

DH1 パケット利用時のスループット

図 7.7 に直接接続した場合と提案システムを用いて接続した場合における Bluetooth A から Bluetooth B への DH1 パケット利用時のスループットを示す. DH1 パケットを用いた ACL リンクの順方向への最大スループットは 172.8Kbps であるので, 1m の近距離で接続した直接接続ではほぼ最大スループットを得ることができている. これに対して, 提案システムを用いて接続した場合は, 最大スループットの約 65%程度のスループットしか得ることができていない. このスループットの低下の原因は, DM1 パケットを利用した場合におけるスループットの低下の原因と同じと考えられる. DH1 パケットの場合も DM1 同様に提案システムの処理の最適化を図ることで直接接続を行った場合と同等のスループットを得ることができると考えられる.

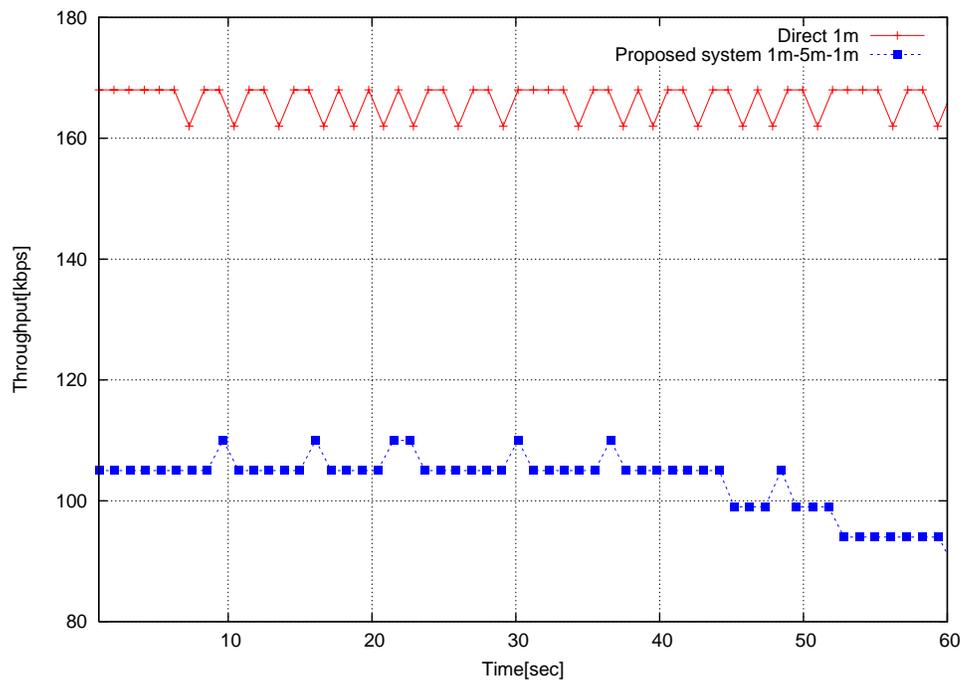


図 7.7: 提案システムのスループット (DH1 パケット)

DM3 パケット利用時のスループット

図 7.8 に直接接続した場合と提案システムを用いて接続した場合における Bluetooth A から Bluetooth B への DM3 パケット利用時のスループットを示す. DM3 パケットを用いた ACL リンクの順方向への最大スループットは 387.2Kbps であるので, 1m の近距離で接続した直接接続では最大スループットに近いスループットを得ることができている. また, 提案システムを用いて接続した場合においても直接接続と同等のスループットを得ることができているこの結果から, DM3 のパケットタイプを利用した場合のスループットに関して提案システムは直接接続と同等の能力を提供できると評価することができる.

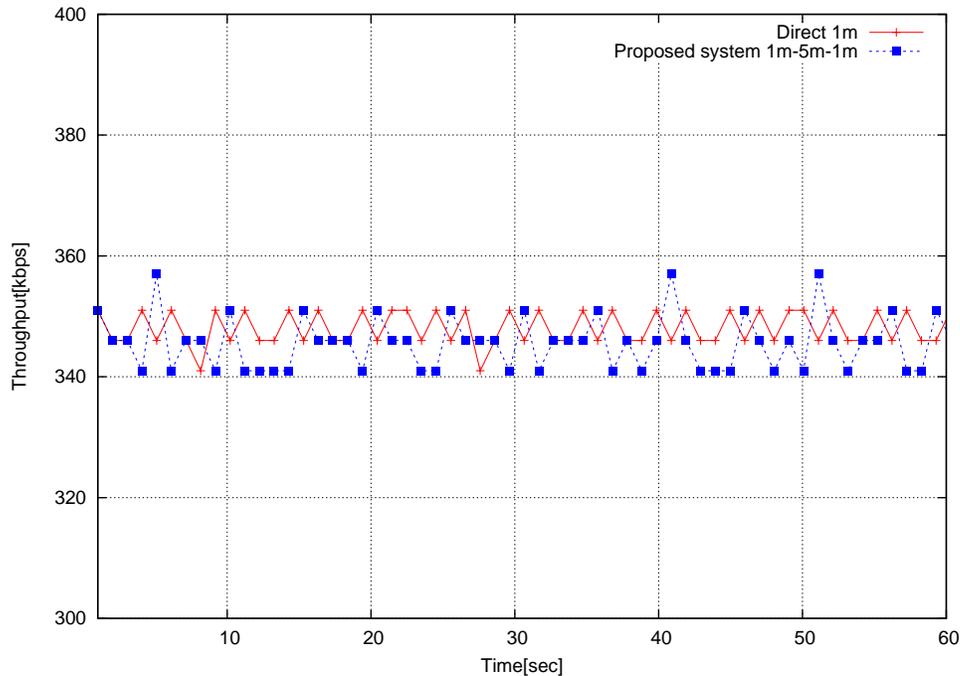


図 7.8: 提案システムのスループット (DM3 パケット使用時)

DH3 パケット利用時のスループット

図 7.9 に直接接続した場合と提案システムを用いて接続した場合における Bluetooth A から Bluetooth B への DH3 パケット利用時のスループットを示す. DH3 パケットを用いた ACL リンクの順方向への最大スループットは 585.6Kbps であるので, 1m の近距離で接続した直接接続では最大スループットに近いスループットを得ることができている. また, 提案システムを用いて接続した場合においても直接接続と同等のスループットを得ることができている. この結果から, DH5 のパケットタイプを利用した場合のスループットに関して提案システムは直接接続と同等の能力を提供できると評価することができる.

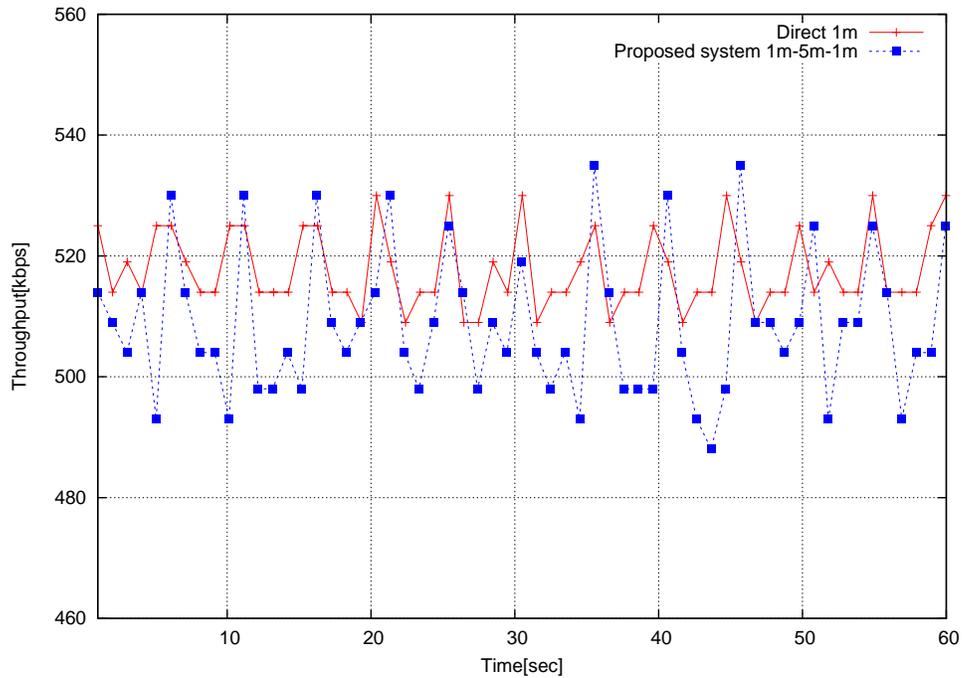


図 7.9: 提案システムのスループット (DH3 パケット使用時)

DM5 パケット利用時のスループット

図 7.10 に直接接続した場合と提案システムを用いて接続した場合における Bluetooth A から Bluetooth B への DM5 パケット利用時のスループットを示す. DM5 パケットを用いた ACL リンクの順方向への最大スループットは 477.8Kbps であるので, 1m の近距離で接続した直接接続では最大スループットに対して約 75%のスループットを得ることができている. これは, 評価のために利用した Bluetooth A と Bluetooth B に接続した HAGIWARA SYS-COM Bluetooth USB Stick とドライバとして利用した Linux BlueZ を用いた場合における DM5 パケットタイプ使用時の性能上の最大値である. この環境において, 提案システムを用いて接続した場合においても直接接続と同等のスループットを得ることができている. この結果から, DM5 のパケットタイプを利用した場合のスループットに関して提案システムは直接接続と同等の能力を提供できると評価することができる.

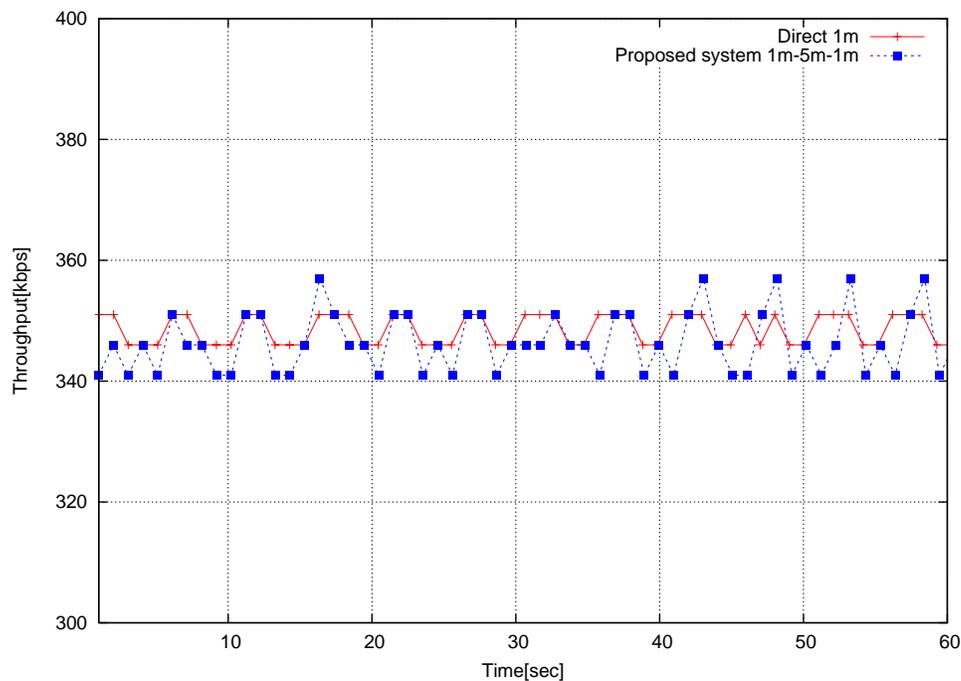


図 7.10: 提案システムのスループット (DM5 パケット使用時)

DH5 パケット利用時のスループット

図 7.11 に直接接続した場合と提案システムを用いて接続した場合における Bluetooth A から Bluetooth B への DH5 パケット利用時のスループットを示す。DH5 パケットを用いた ACL リンクの順方向への最大スループットは 723.2Kbps 75%のスループットを得ることができている。これは、評価のために利用した Bluetooth A と Bluetooth B に接続した HAGIWARA SYS-COM Bluetooth USB Stick とドライバとして利用した Linux BlueZ を用いた場合における DM5 パケットタイプ使用時の性能上の最大値である。この環境において、提案システムを用いて接続した場合においても直接接続と同等のスループットを得ることができている。この結果から、DH5 のパケットタイプを利用した場合のスループットに関して提案システムは直接接続と同等の能力を提供できると評価することができる。

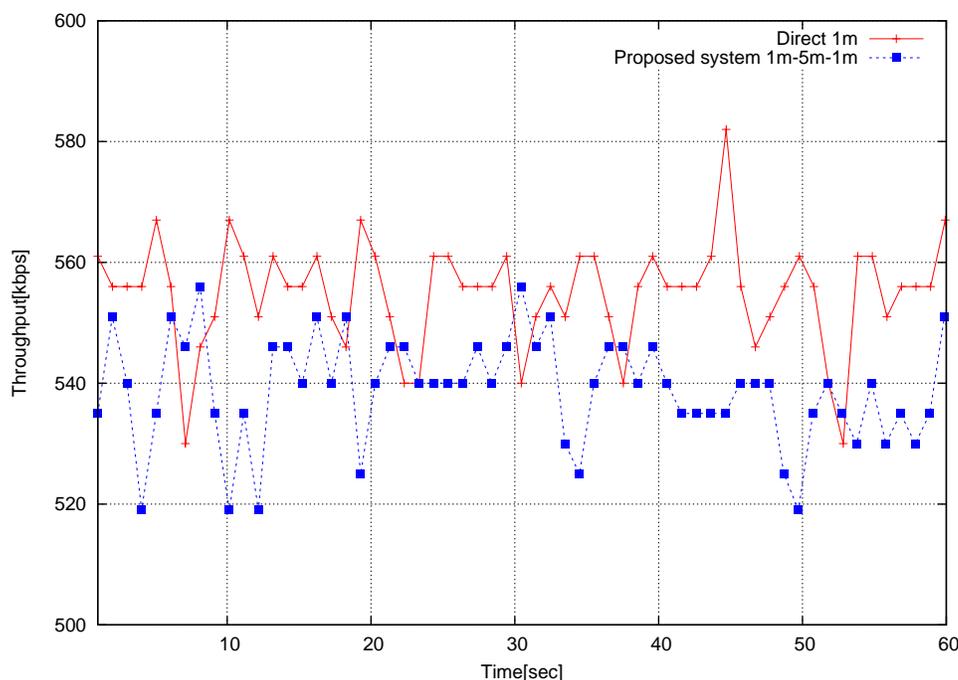


図 7.11: 提案システムのスループット (DH5 パケット使用時)

7.3.4 実装した提案システムに関する考察

本章では、提案する有線接続システムの実装を行い、提案システムを用いて既存の Bluetooth 機器に対する透過接続が可能であることを明らかにした。また、実装した提案システムを用いて Bluetooth 機器間を接続した場合と直接 Bluetooth 機器間の比較をコネクション確立までの時間、伝送遅延、スループットの点から比較を行った。コネクション確立ま

での時間に関して、提案システムを用いることで発生する遅延が接続タイムアウトを発生させる可能性が比較的低いことを示すことができた。また、伝送遅延は、提案システムを用いた場合 10msec から 15msec と非常に小さく、上位プロファイルやアプリケーションに与える影響は少ないことを示すことができた。スループットに関しては、提案システムを用いて直接接続した場合と同等のスループットを提供できることを実装したシステムを用いて確認することができた。これらのことから、提案システムを用いて既存の Bluetooth 機器を透過接続した場合に、提案システムが既存の Bluetooth 機器に与える影響は非常に少ないと評価することができる。

第8章 提案システムの評価

本章では、提案システムの有効性をスループットの点から評価する。直接接続した Bluetooth 機器と提案システムを用いて接続した Bluetooth 機器間のスループットの比較を行う。また、電子レンジを Bluetooth 機器の付近で動作させた Bluetooth 機器にとって干渉の大きい場所における提案システムの有効性を機器間のスループットをもとに評価を行う。

8.1 距離に対する提案システムの有効性

接続された Bluetooth 機器間に距離がある場合、伝送信号の減衰や干渉のために Bluetooth 機器間で安定した通信を行うことができない可能性がある。このような問題に対して、提案システムを用いて機器間の距離を有線伝送路で補うことで Bluetooth 機器間の安定した通信を提供することが可能であると考えられる。この問題に対する提案システムの有効性を検証するために、表 8.1 に示すように、Bluetooth 機器間を直接接続した場合を 4 通り、提案システムを用いた場合の計 5 通りのスループットを各パケットタイプ別に測定した。提案システムを用いて接続した場合におけるスループットの測定は、ローカル側とリモート側の通信リンクで用いるパケットタイプは設定により予め一致させて行った。また、図 8.1 に示すように、評価に利用する Bluetooth 機器および提案システムは、前章における実装環境および実装した提案システム、評価ツールを用いた。

接続方法	接続距離
直接	1m
直接	3m
直接	5m
直接	7m
提案システムによる有線接続	Bluetooth インターフェース間 1m Bluetooth 有線伝送路 5m

表 8.1: 距離に対する提案システムの評価

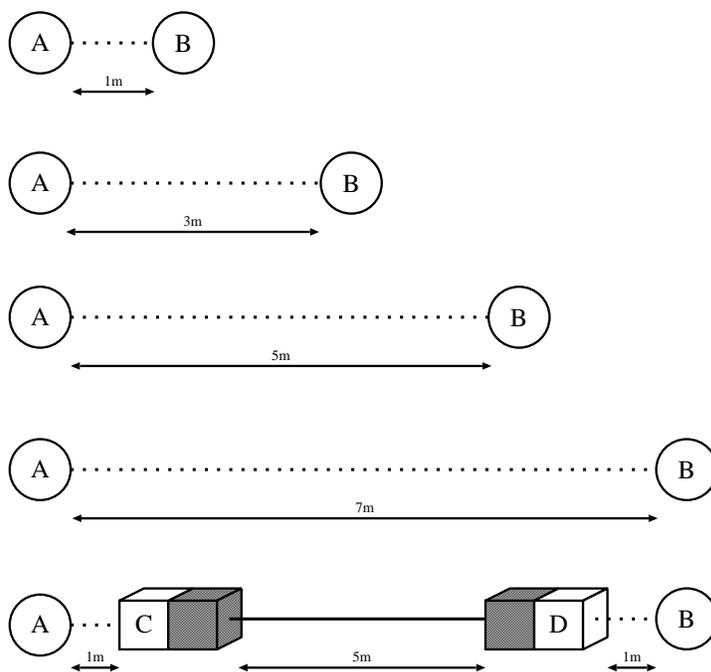


図 8.1: 距離に対する提案システムの評価環境

図 8.2 から図 8.2 に, DM1 から DH5 パケット使用時における各場合のスループットを示す. いずれの場合においても, Bluetooth 機器間を 1m の距離で接続した場合に最も高いスループットを測定され, Bluetooth 機器間の距離が開くほどスループットが低下していることがわかる. とくに最も距離が離れた 7m の距離で接続した場合, スループットが非常に不安定に変動していることから, 無線上でデータの消失が頻繁に起こっていることがわかる. これらの測定結果から, Bluetooth 機器間のスループットは, 機器間の距離に強く依存していることがわかる. これに対して, 提案システムを用いて接続した場合は, 1m の距離で接続した場合と同等のスループットを得ることができるため, Bluetooth 機器間の距離に対するスループットの低下を補うことができている. DM1 パケットを利用した図 8.2 および DH1 パケットを利用した図 8.3 では, 提案システムを用いて接続した場合のスループットは 5m 離れた距離で直接接続した場合に近いスループットが測定されているがこれは前章で説明したとおり提案システムの高速度化により 1m の距離で直接接続した場合と同等なスループットを得ることが可能であると考えられる.

測定の結果より, Bluetooth 機器間のスループットが接続機器間の距離に大きく依存することがわかり, また, 提案システムを用いることでこれらを補うことができることが明らかとなった. 評価を行う際には, Bluetooth 機器間には障害物がない状態で計測したが, 実際の家庭内においては Bluetooth 機器間には様々な障害物が存在する可能性が高い. このような点から, 提案システムが家庭内の Bluetooth 機器間を接続する際に有効であることがいえる. また, 提案システムの有線伝送路は 5m が限界ではなく Bluetooth の接続範

圏外まで延長することが可能であり、かつ、それらの機器に近距離で直接接続している場合と同等のスループットを提供することが可能である。

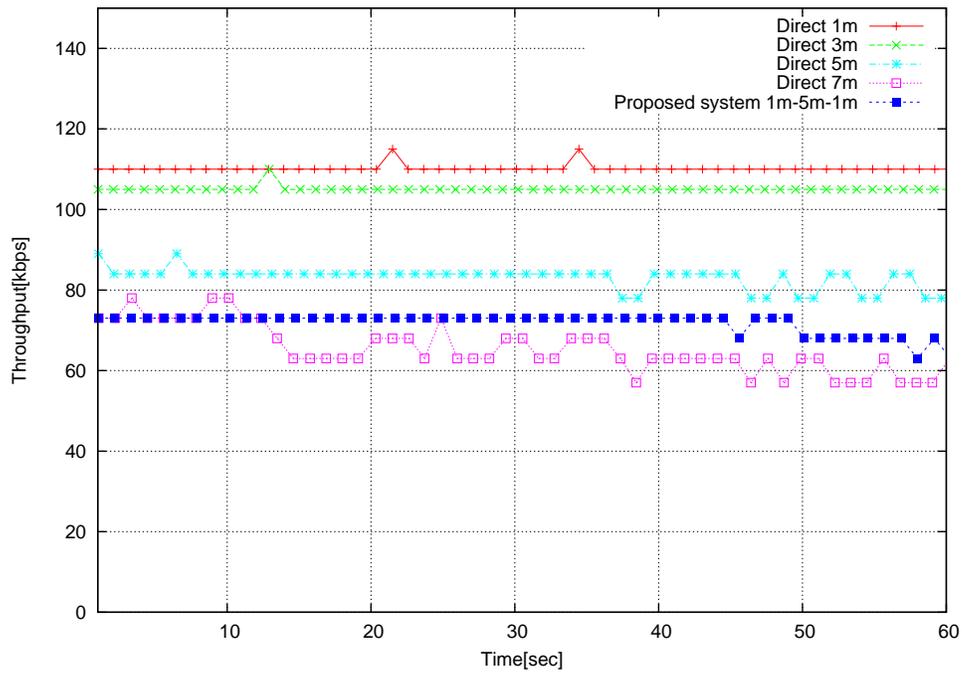


図 8.2: 接続距離に対するスループットの比較 (DM1 パケット使用時)

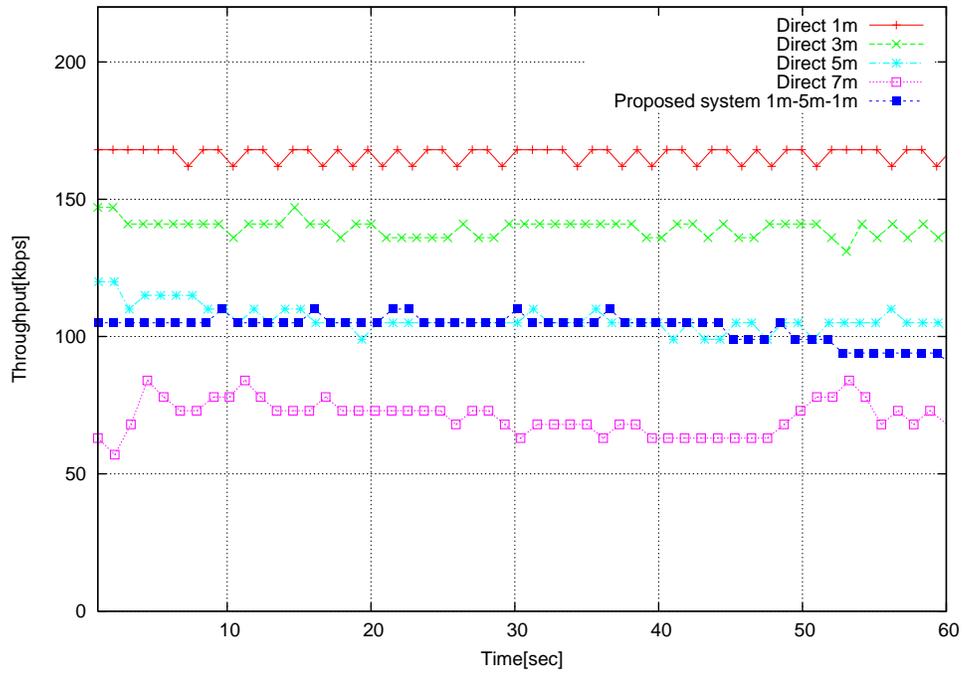


図 8.3: 接続距離に対するスループットの比較 (DH1 パケット使用時)

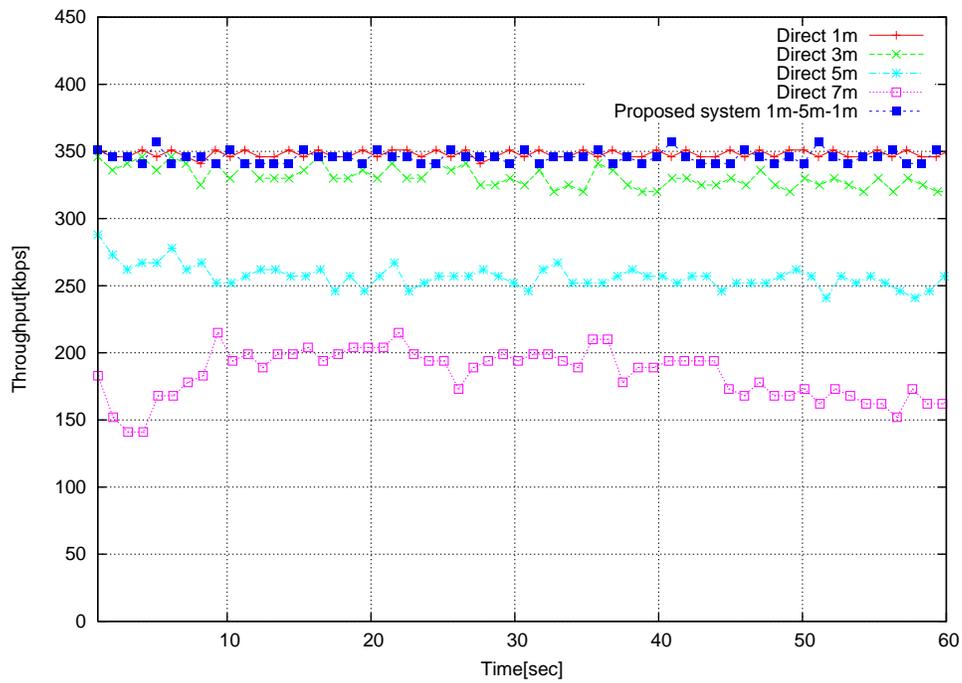


図 8.4: 接続距離に対するスループットの比較 (DM3 パケット使用時)

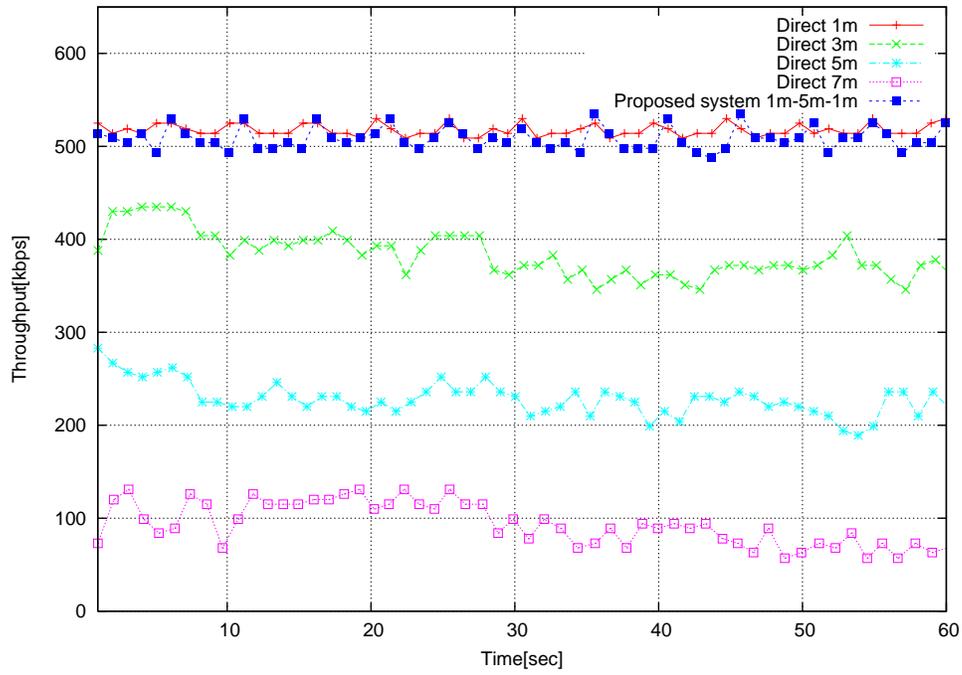


図 8.5: 接続距離に対するスループットの比較 (DH3 パケット使用時)

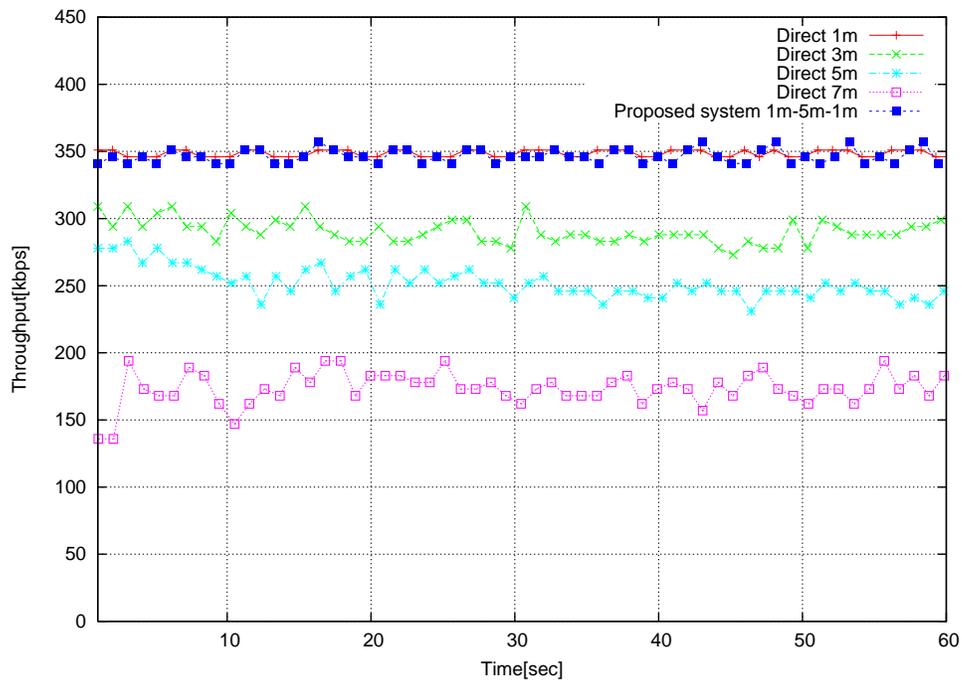


図 8.6: 接続距離に対するスループットの比較 (DM5 パケット使用時)

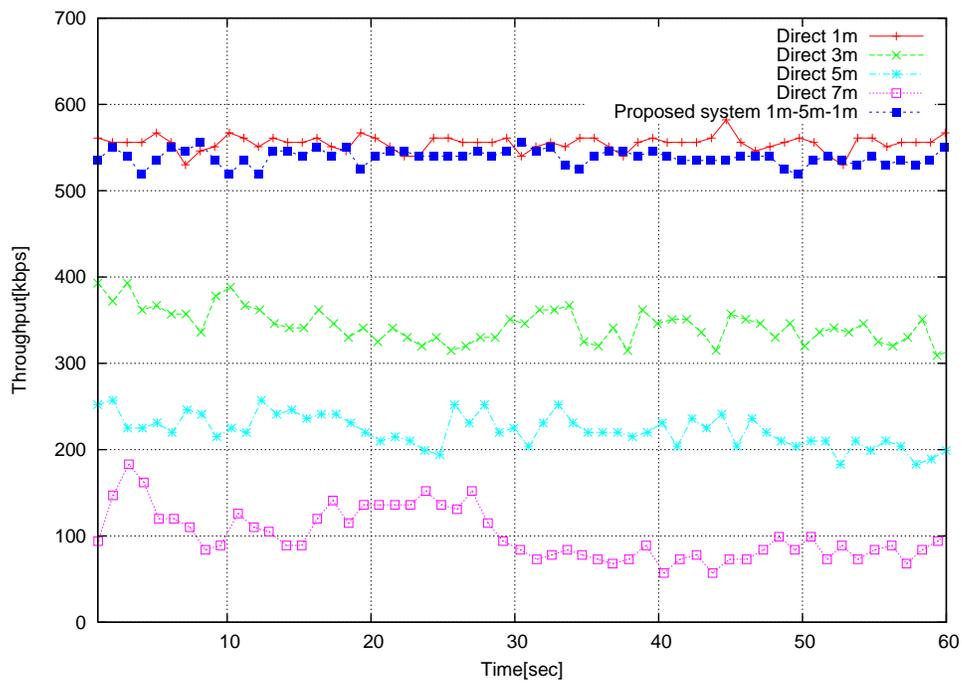


図 8.7: 接続距離に対するスループットの比較 (DH5 パケット使用時)

8.2 干渉に対する提案システムの有効性

家庭内において、Bluetooth 機器を利用する場合、家庭内に存在する電子レンジや無線 LAN など Bluetooth と同一の周波数帯を利用する機器から干渉を受け、安定した通信の提供を妨げられる恐れがある。このような干渉に関する問題に対して、本研究で提案する有線接続システムを用いることで、干渉を受ける場所に存在する Bluetooth 機器間を干渉地帯を有線伝送路を用いて回避することが可能であると考えられる。

干渉に関する問題に対する提案システムの有効性を検証するために、表 8.2 および図 8.2 に示す構成でスループットの測定を行った。測定では、Bluetooth 機器間を 7m の距離で直接接続および提案システムによる接続を行い、Bluetooth 機器間の中間点である 3.5m の地点に干渉源として電子レンジを配置し動作させた。電子レンジを Bluetooth 機器間の直線上に配置していないのは、電子レンジ自体が Bluetooth の伝送信号に対する障害物となることを避けるためである。計測を行う際に利用した電子レンジは次の製品である。

- National オープンレンジ NE-N4
 外寸法 幅 450mm, 奥行き 333mm, 高さ 280mm
 定格電圧電圧 100V, 定格消費電力 1.14kW, 定格高周波出力 600 W

また、前節同様に提案システムを用いて接続した場合における測定の際には、ローカル側とリモート側の通信リンクで用いるパケットタイプは設定により予め一致させて行い、図 8.8 に示すように、評価に利用する Bluetooth 機器および提案システムは、前章における実装環境および実装した提案システム、評価ツールを用いた。

接続方法	接続距離	干渉
直接	7m	なし
直接	7m	あり
提案システムによる有線接続	Bluetooth インターフェース間 1m Bluetooth 有線伝送路 5m	あり

表 8.2: 干渉に対する提案システムの評価

図 8.9 から図 8.9 に、干渉のない環境において 7m の距離で直接接続した場合、電子レンジからの干渉を受ける環境下で 7m の距離で直接接続した場合、同様に干渉を受ける環境下で提案システムを用いて接続した場合において DM1 から DH5 パケットを使用した際のスループットを示す。

いづれの計測結果からも、電子レンジから干渉を受ける環境下において直接接続した場合は、干渉を受けない環境下において直接接続した場合と比較してスループットが低下していることがわかる。これに対して、提案システムを用いて接続した場合は、どちらの環境下において 7m の距離で接続した場合よりも高いスループットを提供できていることが確認できる。これは提案システムを用いることで、7m の距離で直接接続した場合と比べ電

子レンジから離れた場所で無線リンクをそれぞれ確立するため干渉の影響が少ないことに加えて、提案システムの Bluetooth インターフェースと Bluetooth 機器間の距離が近いので Bluetooth 伝送信号の距離による減衰が抑えられたためであると考えられる。測定結果から提案システムによって、家庭内の干渉を受ける場所における Bluetooth 機器間の安定したリンクを提供できることが確認できた。

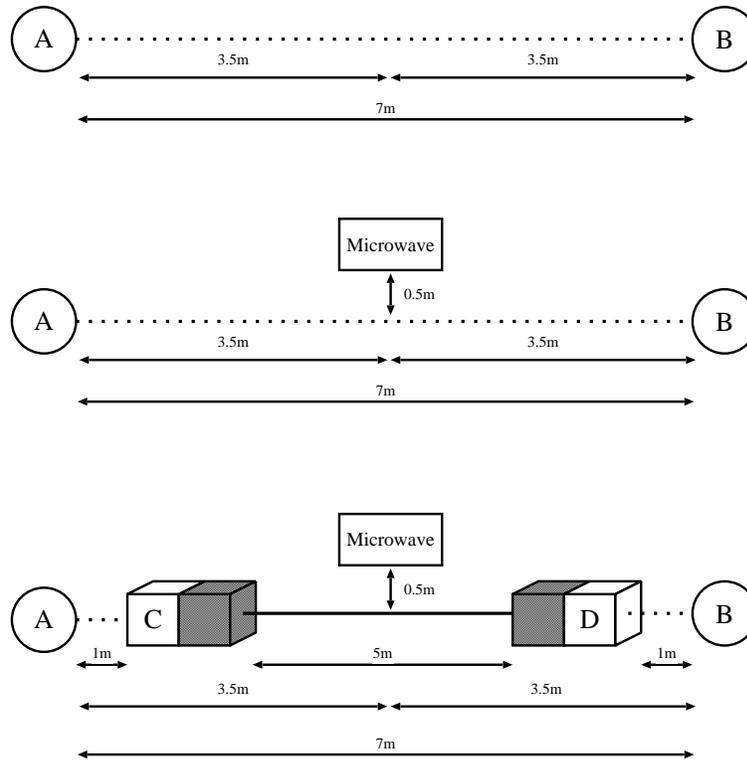


図 8.8: 干渉に対する提案システムの評価環境

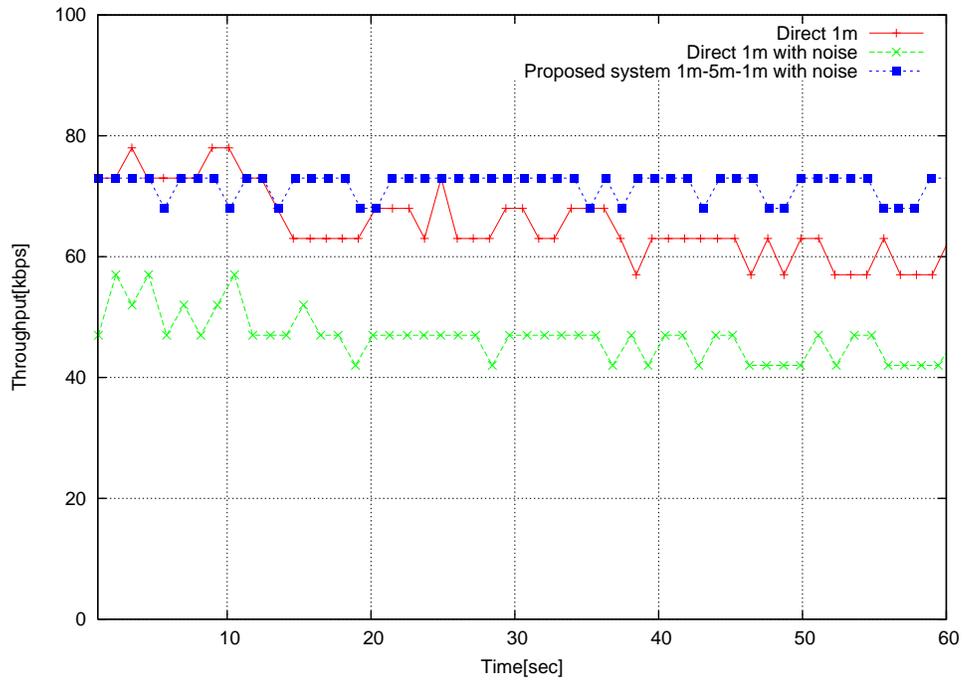


図 8.9: 干渉を受ける環境におけるスループットの比較 (DM1 パケット使用時)

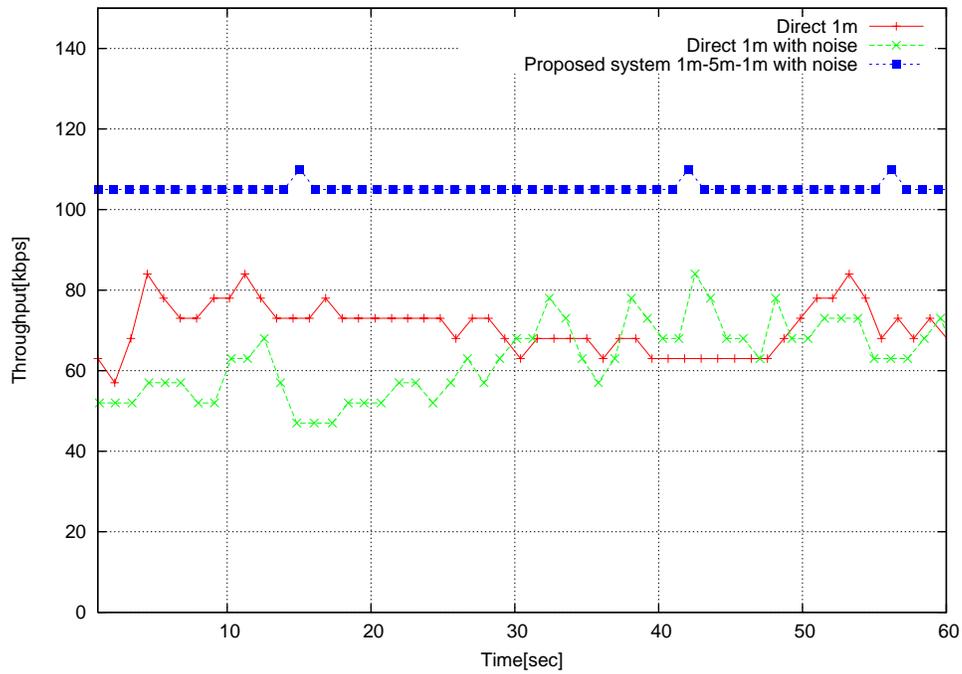


図 8.10: 干渉を受ける環境におけるスループットの比較 (DH1 パケット使用時)

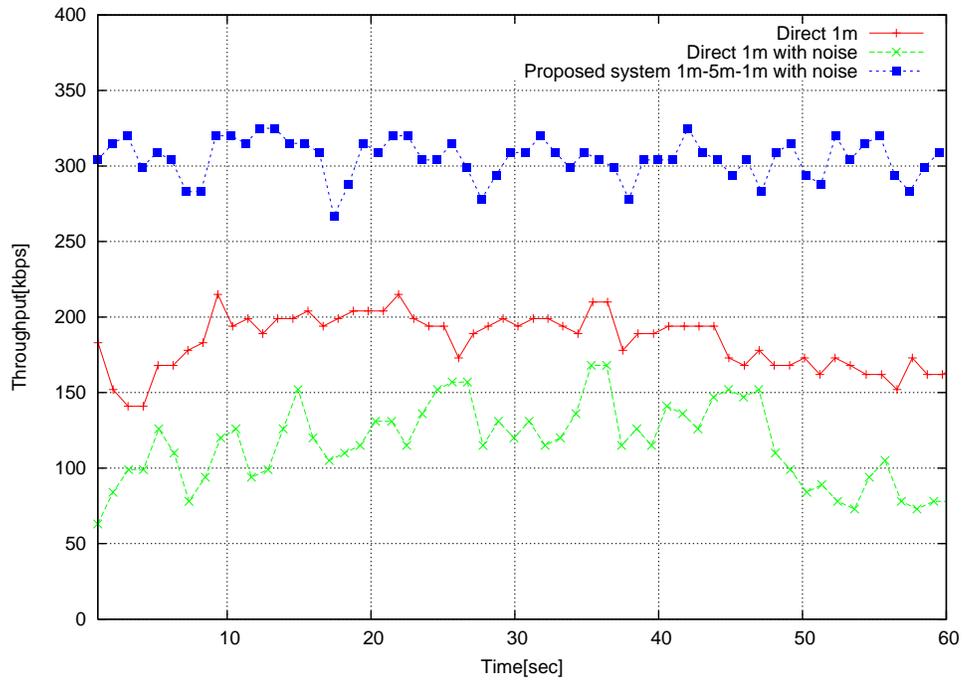


図 8.11: 干渉を受ける環境におけるスループットの比較 (DM3 パケット使用時)

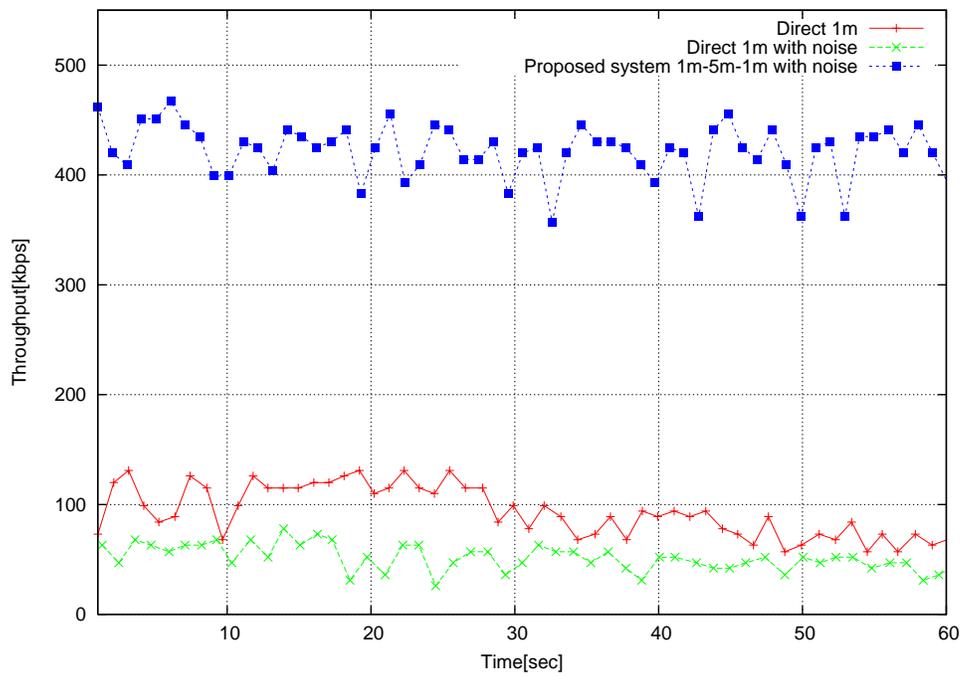


図 8.12: 干渉を受ける環境におけるスループットの比較 (DH3 パケット使用時)

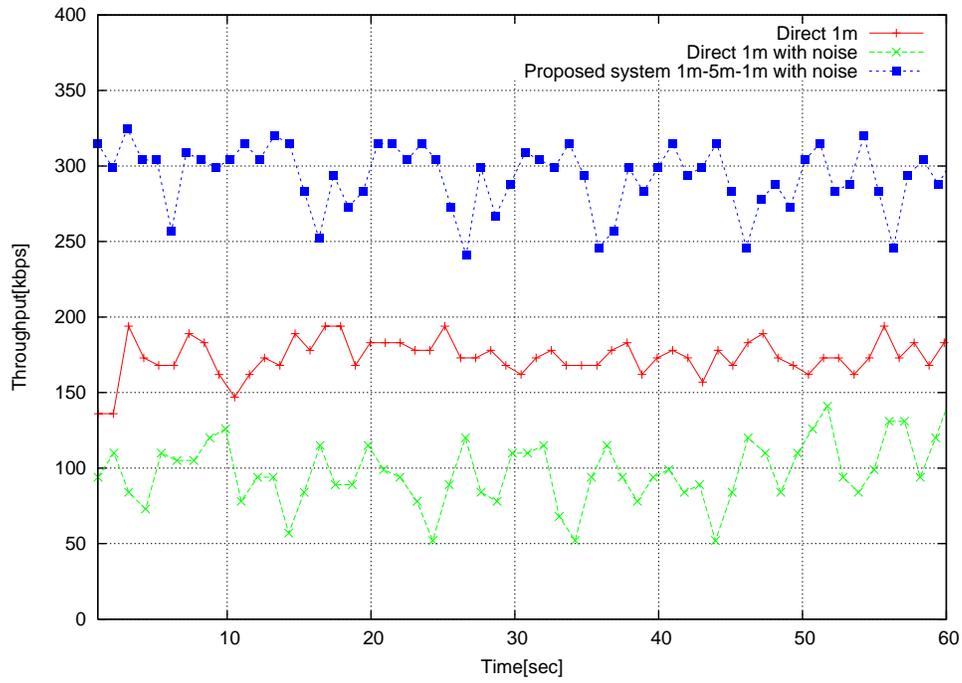


図 8.13: 干渉を受ける環境におけるスループットの比較 (DM5 パケット使用時)

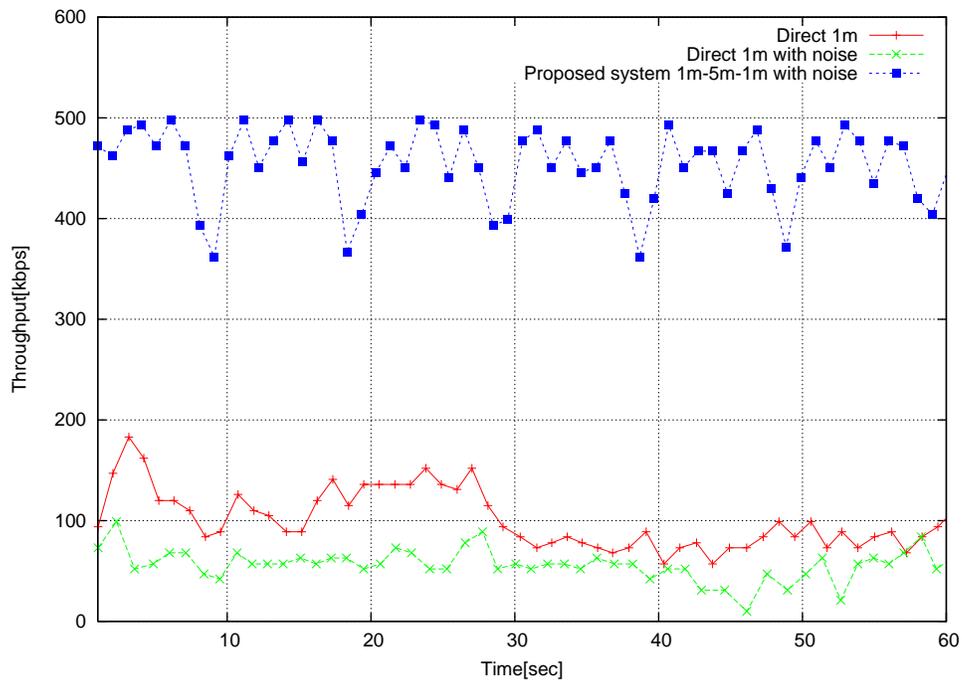


図 8.14: 干渉を受ける環境におけるスループットの比較 (DH5 パケット使用時)

8.3 通信リンクに関するパラメータによる評価

離れた Bluetooth 機器間や干渉の多い場所において提案システムを用いることで直接接続に比べ安定したスループットを得ることができるのは、直接接続に比べて近距離で Bluetooth 機器と提案システムの Bluetooth インターフェース間で通信リンクを確立できるためである。そこで、2つの Bluetooth 機器間の通信リンクに対する Link_Quality および Transmit_Power_Level パラメータの測定を行った。

Link_Quality パラメータは、HCI_Get_Link_Quality コマンドを用いることで得られる Bluetooth 機器間の ACL リンクの品質を表すパラメータである。リンクの品質を測定する方法は、Bluetooth モジュール・ベンダーによって異なるため、測定の際に利用した Bluetooth モジュールによって結果は異なるが、同じ Bluetooth モジュールを利用して行った測定結果の比較に対しては十分な指標となると考えられる。Link_Quality パラメータの値は、0 ~ 255 の範囲で大きいほど品質がよいことを表わしている。Transmit_Power_Level は、HCI_Read_Transmit_Power_Level コマンドを用いることで得られる通信リンクで用いている出力電力レベルである。Transmit_Power_Level パラメータの単位は、dBm である。また、これらのほかに通信リンクに関して指標となるパラメータに RSSI (Receiver Signal Strength Indication) が存在するが、測定を行った環境下では常に Golden Receiver Power Range の範囲内、つまり受信状況が良いことを表す 0 を返したことから評価の対象としていない。

Link_Quality および Transmit_Power_Level パラメータの計測に関しては、これまで提案システムに対して接続する Bluetooth のもつ Bluetooth モジュールとして HAGIWARA SYS-COM Bluetooth USB Stick を利用してきたが、Bluetooth モジュールの返すパラメータ値が詳細なことから、3Com Bluetooth USB Adapter を利用した。ゆえに、測定結果は、3Com Bluetooth USB Adapter に依存した測定結果である。

測定は、図 8.15 に示す配置で Bluetooth 機器間の距離 N を 1 ~ 7m の距離で測定した。表 8.3 に測定結果を示す。

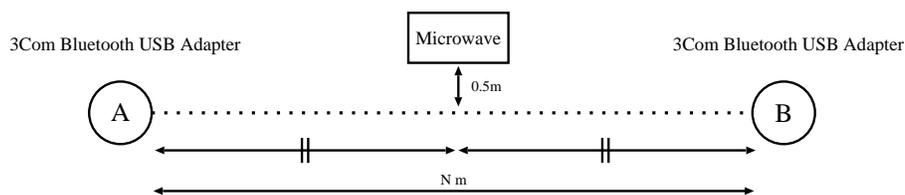


図 8.15: 通信リンクに関するパラメータの測定環境

測定結果より、機器間の距離が大きくなると Link_Quality を維持するために Bluetooth モジュールが出力電力レベルを上げていることがわかる。加えて、干渉が多い場合は Link_Quality パラメータが低下し、これを維持するために出力電力レベルが増加していることがわかる。これまでのスループットの計測結果から通信リンクのスループット値など

	干渉	Link_Quality	Transmit_Power_Level
N=1	なし	255	-12 dBm
N=3	なし	255	-12 dBm
N=5	なし	255	-4 ~ -8 dBm
N=7	なし	255	0 ~ -4 dBm
N=1	あり	213	-12 dBm
N=3	あり	213	-8 dBm
N=5	あり	213	0 ~ -4 dBm
N=7	あり	213	0 ~ -4 dBm

表 8.3: 通信リンクに関するパラメータの測定結果

に関しては、出力電力レベルに関係が大きく関係していることがわかる。また、提案提案システムは、表 8.3 を例にすると 1m の距離で Bluetooth インターフェースと接続した場合は、干渉が存在する場所においても -12dBm の電力レベルで通信を行うことができる。ことから提案システムの有効性を Link_Quality および Transmmit_Power_Level パラメータの面からも提案システムの有効性を評価することが出来たといえる。

第9章 考察と今後の課題

9.1 提案システムの有効性

評価の結果より、本研究で提案する有線接続システムを用いることで、Bluetooth ネットワークの接続範囲の拡大および干渉を受ける場所や、機器間の距離が離れているために安定した通信を行うことができない環境下においても、機器間の安定した通信を実現可能であることが証明された。また実際に提案システムの実装を行い、評価を行った結果からも、提案システムを用いることで接続される Bluetooth 機器に与える影響が小さいと評価することができ、提案システムによる有線接続手法が非常に有効であると考えられる。提案システムを用いることで、Bluetooth ネットワークを用いたホームネットワークの構築が可能である。

9.2 提案システムの実装に関して

本研究で提案する有線接続システムは、Bluetooth モジュールと有線メディアのインターフェースをもつ機器であればソフトウェアのみで実装することが可能である。Manager は、Bluetooth モジュールに対するホストとして動作するプログラムとなり、容易に実装することが可能である点から非常に低コストで実現できるシステムであるといえる。また、比較的実装の自由度が高いことから提案システムを用いた様々なシステムを構築することが可能である。

9.3 提案システムの用いる有線伝送路に関して

9.3.1 有線伝送路間のプロトコル

本研究では、Bluetooth ネットワークを有線伝送路を利用して接続する手法を提案した。提案する有線接続システムの実用には、有線伝送路間のプロトコルが重要となる。例えば、提案システムでは、リモート側の Bluetooth 機器情報をローカル側の Bluetooth インターフェースに対応させることで機器間の透過接続を実現するが、複数の Manager が存在する場合や Bluetooth インターフェースが少数の場合において、どの Bluetooth 機器の情報をどの Manager のもつ Bluetooth インターフェースに対応させるかを決定する仕組みや

Bluetooth 機器と Bluetooth インターフェースの対応関係を管理できる仕組みが提案システムに必要となる。

また、本研究で提案する有線接続手法を用いることで、Bluetooth 機器間を接続する様々なシステムを実現することが可能と考えられる。有線伝送路間のプロトコルの設計次第で、Bluetooth 機器情報の取得機構を利用することによる家庭内の Bluetooth 機器の一元管理の実現、接続される Bluetooth 機器を管理し、Bluetooth インターフェースとの対応関係を操作することによって家庭内において Bluetooth 機器のモビリティを実現するシステムの構築が可能であると考えられる。

9.3.2 有線伝送路の通信メディア

本研究では、提案システムが用いる有線伝送路の通信メディアについて言及していない。これは要求、環境などに応じて利用するメディアは異なると考えられるためである。本研究においては、ホームネットワークを対象としていることから電力線、電話線などの既存配線を利用した通信メディアを利用することが望ましく、これらの伝送メディアを利用することで新規配線を必要としない Bluetooth 機器によるホームネットワークが実現できると考えられる。

また、伝送路と関連する検討事項として Bluetooth の同期通信チャネルである SCO に関する問題がある。同期通信チャネルを接続する際には、伝送路にも同期型の通信メディアが望まれる、しかし提案システムの利用のために高度な通信メディアが必要となることは好ましくない。非同期型の通信メディアを利用した同期チャネルの接続に関しては、今後の検討事項である。

9.4 Bluetooth 規格に関して

本論文で提案する有線接続システムは、RF から HCI を持つ Bluetooth モジュールを Bluetooth インターフェースとして利用した Bluetooth 規格に準拠したシステムとなっている。提案システムは、HCI イベントを利用し通信リンクに対する情報を取得し、HCI イベントに対応する HCI コマンドを用いることで Bluetooth 機器間の透過接続を実現した。しかし、Baseband 層や LM 層の情報の中には HCI イベントから得られない情報があり、得られない情報に関してはそれを補う機構が提案システムに必要であった。これらの機構のいくつかは、Bluetooth 規格に準拠しない特殊なデバイスを設計することで解決することが可能であると考えられる。以下に挙げる課題に関しては、特に特殊なデバイスを用いて解決することが望まれ、また、これらの課題を解決することでより実用的な Bluetooth ネットワークの有線接続システムを実現することが可能となる。

9.4.1 ピコネット内同期

提案システムは、ローカル側とリモート側でそれぞれ異なるピコネットを形成する。これによって、前述の通り Baseband 層における通信遅延許容時間に対する問題を回避することが可能となる。しかし、一方のピコネットにおけるピコネット内同期のタイミングをもう一方で得ることが出来ないという問題があった。この問題にたいして、Baseband 層において Inquiry リクエストを受けたことを Manager へ通知する機構が存在したならば、比較的容易に双方のピコネット内同期のタイミングを一致させることが可能となる。また、これにより必要な場合にのみ Inquiry を行うことが可能となり定期的に Inquiry を行うことがなくなるため、Inquiry による干渉の発生を軽減することが可能となる。

9.4.2 リモート側の Bluetooth 機器情報の取得

提案システムは、リモート側の Bluetooth 機器の情報を予め収集し、ローカル側の Bluetooth インターフェースに情報を反映させることでリモート側の Bluetooth 機器情報をローカル側へ提供する。この機構に関して、Inquiry リクエストおよび Name リクエストを受けたことを Manager へ通知する機構をもつことが可能であれば、直接リモート側の Bluetooth 機器情報をローカル側へ提供することも可能であると考えられる。Inquiry リクエストに関しては、リクエストに対する返答メッセージである FHS パケットはあるランダム時間待って返信される。これは、FHS パケットの衝突を防ぐためであるが、このリクエストに即答する必要がないことを利用し、その間にリモート側でも Inquiry を行うことも可能であると考えられる。また、Name リクエストに関しては、Bluetooth 機器の名前は HCI イベントとして通知されない特別な ACL リンクを確立して受信する。ACL リンクの確立は、提案システムを用いる手法で可能なため、Name リクエストに対する特別な ACL リンクに関しても同様にあつかうことも可能であれば、直接リモート側の Bluetooth 機器から名前情報を取得することができると考えられる。これらのことから、リモート側の Bluetooth 機器情報の取得に関しても、特殊なデバイス、および Baseband 層、LM 層へ有線接続システムのための機能の追加を行うことで実現できると考えられる。

9.4.3 Bluetooth デバイスアドレス

Bluetooth 規格では、1つの Bluetooth 機器には必ず1つの Bluetooth デバイスアドレスが必要である。これは、Bluetooth 規格では、周波数ホッピングのパターンをマスターの Bluetooth デバイスアドレスを用いて算出し、マスターとスレーブはお互いに算出された周波数ホッピングパターンに同期することで機器間の通信を実現するためである。このため、Bluetooth では通信を行うために必ず Bluetooth デバイスアドレスが必要となり、提案する有線接続システムも自身の Bluetooth デバイスアドレスを持つことが必要であった。加えて、Bluetooth 規格では、1つの Bluetooth デバイスに対して複数の Bluetooth デ

デバイスアドレスを与えることを想定していないため、このため、提案する有線接続システムは接続される Bluetooth 機器の識別のために、複数の Bluetooth デバイスを Bluetooth インターフェースとして持つことが必要であった。結果として、提案システムは、接続される Bluetooth 機器の数だけ Bluetooth インターフェースを持つことが必要となり、接続システムとしての柔軟性に欠けるという問題が生じた。

これに対し、複数の Bluetooth デバイスアドレスを有することができるデバイスを実現することが出来れば、複数の Bluetooth インターフェースを持つ機構は必要となくなり、接続システムの資源の無駄をなくすことが可能となる。また、これによって接続システムとしての柔軟性の向上を図ることが可能となる。複数の Bluetooth デバイスアドレスを有するデバイスは、Baseband 層において Bluetooth デバイスアドレスとホッピングパターンを管理することによって実現可能であると考えられる。1つの Bluetooth アドレスを持つ複数の Bluetooth 機器が通信を行っている際には、異なるピコネットの Bluetooth 機器同士は異なるホッピングパターンを利用し、互いに干渉を発生させないようにしている。同様のことが、複数の Bluetooth デバイスアドレスをもつ1つのデバイスとの通信においても実現可能であると考えられ、また、デバイスが1つであることから、ホッピングにおける利用周波数帯の衝突も予め回避できようなデバイスが実現可能であると考えられる。現在の Bluetooth 規格では、本研究で提案するような有線接続システムを想定していないため複数の Bluetooth デバイスアドレスを持つ Bluetooth デバイスは規定されていないが、このようなデバイスの実現を規格に加えることは、今後、Bluetooth ネットワークを拡張する需要に答えるために必要であると考えられる。

9.4.4 パケットタイプの一致

提案システムでは、HCI イベントをもとにローカル側とリモート側の通信リンクの管理を行うが、通信リンクで実際に利用されているパケットタイプを得ることができないため、利用しているパケットタイプの推定、または利用するパケットタイプを接続対象の Bluetooth 機器に強制する必要がある。この情報を得ることができれば、この問題は容易に解決できる。

9.4.5 理想的な有線接続システム

本論文で提案する有線接続システムの最も問題となる点は、ローカル側の Bluetooth 機器がリモート側の Bluetooth 機器情報を提案システムを用いて間接的に得ていることである。このため、提案システムを用いる場合接続される Bluetooth 機器をある程度、意識的に特定しておく必要がある。しかし、前述するような特殊なデバイスを設計することで機器情報を直接取得する本来の Bluetooth ネットワークの利用方法に近い利用を提案システムを意識せずに行うことが可能となると考えられる。実際に設計を行う際は、Baseband 層におけるデバイスアドレスの管理など、より複雑な処理が必要となり、処理時間に関す

る制約も大きくなると考えられる。しかし、特殊なデバイスによって十分に価値のある有線接続システムを構築できると考えられる。

9.5 他の手法との比較

提案システムによる有線接続手法と物理層による有線接続手法、プロファイルによる有線接続手法との比較を図 9.1 にまとめる。物理層による接続手法の実現には、 $10\mu\text{sec}$ 以内の中継を実現する高度なハードウェア技術が必要であるためコスト面で高く、またプロファイルによる接続手法はコストが低いが、有線接続システムを利用するすべての Bluetooth 機器がプロファイルを持たなくてはならないため最終的にコストが高くなる。提案システムを用いた場合は、Bluetooth 規格に準じた Bluetooth 機器であればすべて接続が可能であり、低コストで実現することが可能である。

また、接続可能な距離に関しては、物理層による有線接続システムは遅延に対する制約が大きいことから近距離の機器間しか接続することはできないと考えられる。遠距離の Bluetooth 機器間の接続には、予め有線接続のために設計されたプロファイルによる有線接続手法が最も適している。提案システムを用いた場合は許容可能な遅延は、L2CAP 層より上位のプロファイルやアプリケーションのパラメータによるが、家庭内のネットワークにおいては十分である。

	コスト	接続距離	接続可能な機器
提案システム	低	近～中	すべて
物理層による接続	高	近	すべて
プロファイルによる接続	低～高	近～遠	プロファイルをもつ機器のみ

表 9.1: 有線接続手法の比較

第10章 まとめ

本研究では, Bluetooth ネットワークを有線拡張する接続システムを提案し, 特別なプロファイルを持たない既存の Bluetooth 機器の透過な有線接続, ノイズや干渉を受ける場所における安定した機器間の接続, Bluetooth 接続範囲の柔軟な拡張を可能にする有線接続システムを設計し, 提案システムの実装および評価を行った.

Bluetooth ネットワークの有線拡張の手法として, 物理層, データリンク層, プロファイルによる接続手法に対してそれぞれ検討を行い, 本研究では, Bluetooth の HCI における HCI イベントおよび HCI データパケットの転送を行うことで既存の Bluetooth 機器の透過接続を実現する有線接続システムを提案した.

実装された提案システムを用いて, 既存の Bluetooth 製品による接続検証を行い, 提案システムが用いる有線接続手法によって既存の Bluetooth 機器間を透過に接続可能であることを証明した. また, Bluetooth 機器を提案システムを用いて有線接続した際に, 提案システムが Bluetooth 機器に与える影響を実装した提案システムを用いて測定し, 提案システムを用いることによるコネクション確立時の遅延は最大で約 4 秒であり, 上位プロトコルのタイムアウト値と比較して十分に影響が少ないことを示した. データの伝送遅延は約 10msec から 15msec と非常に小さく, 有線伝送路の距離をかなり延長したとしても十分に通信が可能であることを示し, 提案システムを介して接続された Bluetooth 機器間が直接接続した場合と同等なスループットを得ることができることを示した. そして, 提案システムの有効性の評価を, Bluetooth 機器の接続距離による通信リンクの安定性に対する問題と干渉を受ける場所における Bluetooth 機器間のリンクの安定性に関して, 提案システムを用いて無線区間を縮めることで無線伝送ロスを軽減し解決出来ることを示した.

本研究で提案する有線接続システムを用いることで, 特別なプロファイルを持たない既存の Bluetooth 機器の透過な有線接続, ノイズや干渉を受ける場所における安定した機器間の接続, Bluetooth 接続範囲の柔軟な拡張が可能となる.

謝辞

本研究を進めるにあたり、研究の方向性や着目点について常に指針を与えてくださり、また研究者としての物事の捉え方の心得を授けて下さった丹康雄助教授に深く感謝致します。また、本研究に関して多くの有意義なご意見を頂きました博士課程の中田潤也氏、研究中に幾度も貴重な助言を頂きました牧野義樹氏に心から感謝致します。

さらに、隣のブースで研究や人生について語った余暁武氏、必死に研究を行う姿勢を教えてくれた名取昭正氏、本研究の実験を手伝ってくれた後輩の水口千賀夫氏、松岡敬生氏に感謝致します。そして、励ましあいながら共に研究生活を過ごした丹研究室の皆様心から感謝致します。

また、石川での研究以外の活動では、金沢大学の藤田研究室の皆様、テニスサークルの Life21 の皆様、大学時代からの友人達には大変お世話になりました。皆様には、心から感謝しております。

最後に、研究生活を様々な面で支えてくれた両親に感謝致します。

参考文献

- [1] Bluetooth, “The Official Bluetooth Website,”
<http://www.bluetooth.com/>
- [2] ECHONET, “ECHONET CONSORTIUM,”
<http://www.echonet.gr.jp/>
- [3] UPnP, “UPnP FORUM,” <http://www.upnp.org/>
- [4] IEEE802.11, “IEEE 802.11, The Working Group for Wireless LANs,”
<http://grouper.ieee.org/groups/802/11/>
- [5] UWB, “Ultra Wideband Working Group,” <http://www.uwb.org>
- [6] 宮津和弘, “Bluetooth 技術解説ガイド,” リック・テレコム, 2001
- [7] IEEE802, “IEEE 802 LAN/MAN Standards Committee,” <http://www.ieee802.org>
- [8] BlueZ, “Official Linux Bluetooth protocol stack,” <http://www.bluez.org/>

付録A 提案システムの関数紹介

提案システムで実装した提案システムのライブラリ関数を示す。

A.1 全体構成

図 A.1 に実装した提案システムとライブラリ関数の全体図を示す。実装した提案システムは、Bluetooth インターフェースと有線伝送路に対して、それぞれパケットを処理するプロセスを起動し、これらのプロセスを介して提供されるイベントパケットやデータパケットをメインプロセスで処理をする構造になっている。

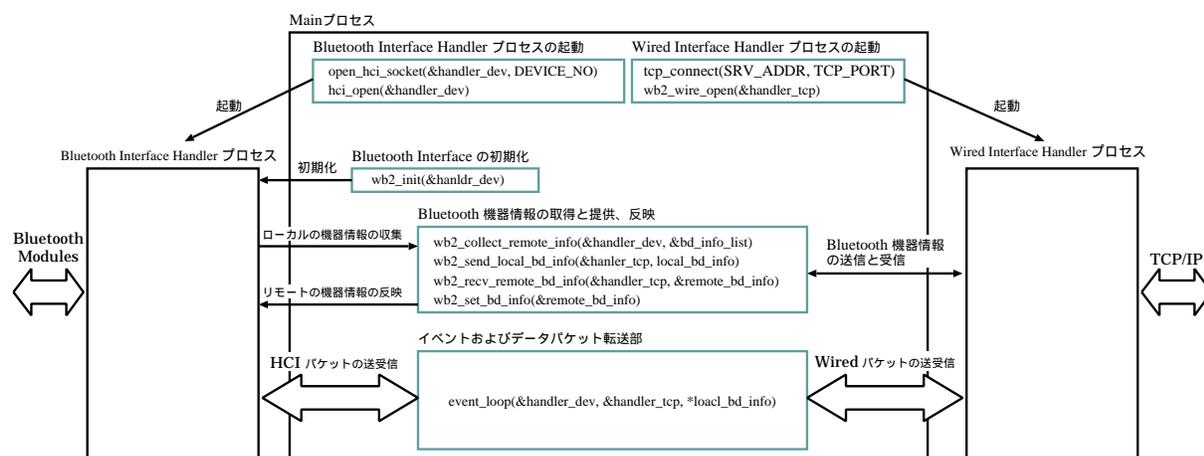


図 A.1: 実装した提案システムの全体図

A.2 Bluetooth Interface Handler

A.2.1 open_hci_socket

書式

```
int open_hci_socket(hci_t *handle, int devid)
```

説明

devid で指定された Bluetooth モジュールに対する HCI socket をオープンし、指定した handle に対応付ける。Bluetooth Interface Handler プロセスを起動する hci_open 関数を呼ぶ前に本関数によって handle と Bluetooth モジュールを対応付ける必要がある。

引数

hci_t *handle

hci_t 構造体は、Bluetooth Interface Handler プロセスおよび Wired Interface Handler プロセスと通信を行うための入出力識別子が格納される構造体であり、各プロセスに対する操作はこのハンドルを指定することで実行可能である。プロセスをハンドルするための構造体 hci_t へのポインタを指定する。

int devid

接続されている Bluetooth デバイスの ID。Linux BlueZ では 0 から始まるデバイス ID が接続された順に割り当てられている。

戻り値

関数が成功した場合、devid で指定された Bluetooth インターフェースに対する HCI ソケットの識別子が返される。

A.2.2 hci_open

書式

```
hci_t *hci_open(hci_t *handle)
```

説明

handle 中の HCI socket の識別子に対する Bluetooth Interface Handler プロセスを起動する。

引数

hci_t *handle

起動したプロセスのハンドルとして対応させたい hci_t 構造体に対するポインタを指定する。

戻り値

関数が成功した場合は、Bluetooth Interface Handler プロセスに対するハンドルへのポインタが返る。

A.2.3 hci_close

書式

```
hci_t hci_close(hci_t *handle)
```

説明

handle で指定された Bluetooth Interface Handler プロセスを停止する.

引数

hci_t *handle

停止したい Bluetooth Interface Handler プロセスに対するハンドルへのポインタを指定する.

戻り値

関数に成功した場合は, 指定したハンドルに対するポインタが返る. 失敗した場合は, NULL が返る.

A.3 Wired Interface Handler

A.3.1 wired_open

書式

```
hci_t wired_open(hci_t *handle)
```

説明

handle 中の TCP ソケット識別子に対する Wired Interface Handler プロセスを起動する.

引数

hci_t *handle

起動したプロセスのハンドルとして対応させたい hci_t 構造体に対するポインタを指定する.

戻り値

関数が成功した場合は, Wired Interface Handler プロセスに対するハンドルへのポインタが返る.

A.3.2 wired_close

書式

```
hci_t wired_close(hci_t *handle)
```

説明

handle に対応する Wired Interface Handler プロセスを停止する.

引数

`hci_t *handle`

停止したい Wired Interface Handler プロセスに対するハンドルへのポインタを指定する.

戻り値

関数に成功した場合は, 指定したハンドルに対するポインタが返る. 失敗した場合は, NULL が返る.

A.4 Manager

A.4.1 wb2_init

書式

```
void wb2_init(hci_t *handle)
```

説明

handle に対応する Bluetooth インターフェースの初期化を行い, Bluetooth インターフェースを接続要求受付状態にする.

引数

初期化を行いたい Bluetooth インターフェースのハンドルへのポインタを指定する.

戻り値

なし.

A.4.2 wb2_collect_remote_bd_info

書式

```
int wb2_collect_remote_bd_info(hci_t *handle, BD_INFO_LIST *list)
```

説明

handle に対応する Bluetooth インターフェースを用いて接続範囲に存在する Bluetooth 機器の検索を行い, 発見した Bluetooth 機器の情報を収集する. 収集した情報を Bluetooth デバイスアドレスごとにリスト化したものを list に返す.

引数

hci_t *handle

情報収集フェーズを行わせる Bluetooth インターフェースに対応するハンドルへのポインタを指定する.

BD_INFO_LIST * list

情報取得フェーズを行った結果, 得られた情報を BD_INFO のリスト構造で提供されるリスト list へのポインタを指定する.

戻り値

関数が成功した場合に 0 が返り, list で指定されるリストには取得した周辺の Bluetooth 機器の情報が与えられる. Bluetooth 機器が発見できなかった場合は, リストには何も追加されない.

A.4.3 int wb2_send_local_bd_info

書式

```
int wb2_send_local_bd_info(hci_t *handle, BD_INFO *bd_info)
```

説明

handle に対応する有線伝送路に対して, ローカル側で収集した周囲の Bluetooth 機器の情報 bd_info を送信する.

引数

BD_INFO *bd_info

有線伝送路を介して送信したい Bluetooth 機器の情報 BD_INFO 構造体へのポインタを指定する.

戻り値

関数が成功した場合は正の数が返り, 失敗した場合は-1 が返る.

A.4.4 wb2_rcv_remote_bd_info

書式

```
int wb2_rcv_remote_bd_info(hci_t *handle, BD_INFO *bd_info)
```

説明

handle に対応する有線伝送路から, リモートの Manager の周囲に存在する Bluetooth 機器の情報 bd_info を受信する.

引数

hci_t *handle

Bluetooth 機器情報を受信したい有線伝送路に対応するハンドルへのポインタを指定する.

BD_INFO *bd_info

受信した Bluetooth 機器情報を格納するための BD_INFO 構造体へのポインタを指定する.

戻り値

関数が成功した場合は正の数が返り, 失敗した場合は-1 が返る.

A.4.5 wb2_set_bd_info

書式

```
void wb2_set_bd_info(hci_t *handle, BD_INFO *bd_info)
```

説明

handle に対応する Bluetooth インターフェースに対してリモートリモートの Manager から得た Bluetooth 機器の情報 bd_info を反映させる. Bluetooth インターフェースの名前, CoD が bd_info で指定された内容に更新される.

引数

hci_t *handle

Bluetooth 機器の情報を反映させたい Bluetooth インターフェースに対応するハンドルへのポインタを指定する.

BD_INFO *bd_info

Bluetooth インターフェースへ反映させたい Bluetooth 機器情報へのポインタを指定する.

戻り値

なし.

A.4.6 wb2_event_loop

書式

```
int wb2_event_loop(hci_t *handle_dev, hci_t *handle_wired, BD_INFO *bd_info)
```

説明

接続された Bluetooth インターフェースまたは有線伝送路から接続要求を受けた場合に, 対応する Bluetooth インターフェースのハンドル handle_dev と有線伝送路のハンドル handle_wired を指定することで HCI イベントおよび HCI データパケッ

トの転送を行う。有線伝送路からの接続要求の場合には, bd_info に提案システムから接続を要求する Bluetooth 機器の情報を与える。

引数

hci_t *handle_dev

転送処理を行う Bluetooth インターフェースに対応するハンドルへのポインタを指定する。

hci_t *handle_wired

転送処理を行う有線伝送路に対応するハンドルへのポインタを指定する。

BD_INFO *bd_info

提案システムから接続要求を行う Bluetooth 機器に対する情報へのポインタを指定する。

戻り値

関数が成功した場合に 1 が返る。

A.4.7 send_hci_cmd_pkt

書式

```
int send_hci_evt_pkt(hci_t *handle, hci_cmd_pkt *cmd)
```

説明

handle に対応する Bluetooth インターフェースに対して HCI コマンドパケットを送信する。

引数

hci_t *handle

HCI コマンドパケットを送信したい Bluetooth インターフェースに対応するハンドルへのポインタを指定する。

hci_cmd_pkt *cmd

送信したい HCI コマンドパケットへのポインタを指定する。

戻り値

関数が成功した場合は, 送信したパケットのデータ数が返る。失敗した場合は-1 が返る。

A.4.8 recv_hci_evt_pkt

書式

```
int recv_hci_evt_pkt(hci_t *handle)
```

説明

handle に対応する Bluetooth インターフェースから HCI イベントパケットを受信する.

引数

hci_t *handle

HCI イベントパケットを受信したい Bluetooth インターフェースに対応するハンドルへのポインタを指定する.

戻り値

関数が成功した場合は, 受信したパケットのデータ数が返る. 失敗した場合は-1 が返る.

A.4.9 send_hci_acl_pkt

書式

```
int send_hci_acl_pkt(hci_t *handle, hci_acl_pkt *acl)
```

説明

handle に対応する Bluetooth インターフェースまたは有線伝送路に対して HCI ACL データパケットを送信する.

引数

hci_t *handle

HCI ACL データパケットを送信したい Bluetooth インターフェースに対応するハンドルへのポインタを指定する.

hci_acl_pkt *acl

送信したい HCI ACL データパケットへのポインタを指定する.

戻り値

関数が成功した場合は, 送信したパケットのデータ数が返る. 失敗した場合は-1 が返る.

A.4.10 recv_hci_acl_pkt

書式

```
int recv_hci_acl_pkt(hci_t *handle)
```

説明

handle に対応する Bluetooth インターフェースまたは有線伝送路から HCI ACL データパケットを受信する.

引数

`hci_t *handle`

HCI ACL データパケットを受信したい Bluetooth インターフェースに対応するハンドルへのポインタを指定する.

戻り値

関数が成功した場合は, 受信したパケットのデータ数が返る. 失敗した場合は-1 が返る.

A.4.11 `send_hci_sco_pkt`

書式

```
int send_hci_sco_pkt(hci_t *handle, hci_sco_pkt *sco)
```

説明

`handle` に対応する Bluetooth インターフェースまたは有線伝送路に対して HCI SCO データパケットを送信する.

引数

`hci_t *handle`

HCI SCO データパケットを送信したい Bluetooth インターフェースに対応するハンドルへのポインタを指定する.

`hci_sco_pkt *sco`

送信したい HCI SCO データパケットへのポインタを指定する.

戻り値

関数が成功した場合は, 送信したパケットのデータ数が返る. 失敗した場合は-1 が返る.

A.4.12 `recv_hci_sco_pkt`

書式

```
int recv_hci_sco_pkt(hci_t *handle)
```

説明

`handle` に対応する Bluetooth インターフェースまたは有線伝送路から HCI SCO データパケットを受信する.

引数

hci_t *handle

HCI SCO データパケットを受信したい Bluetooth インターフェースに対応するハンドルへのポインタを指定する.

戻り値

関数が成功した場合は, 受信したパケットのデータ数が返る. 失敗した場合は-1 が返る.