Power Saving Approaches for use with Bluetooth:
Modelling and Analysis of new Protocols

Jiangchuan Wen, MEng

A thesis submitted for the degree of
Doctor of Philosophy

Supervisor
Dr. John Nelson

Wireless Access Research Centre
Department of Electronic and Computer Engineering
Faculty of Science and Engineering
University of Limerick

2012
Declaration

The substance of this thesis is the original work of Jiangchuan Wen, and due reference and acknowledgement has been made, where necessary, to the work of others. No part of this thesis has been submitted to any other university or higher education institution, or for any other academic award in this University.

Jiangchuan Wen (PhD student)

Dr. John Nelson (Supervisor)

This thesis was defended on the ___ of May, 2012

Examination Committee

Chairman          Dr. Ciaran MacNamee          University of Limerick, Ireland
External Examiner Prof. Pietro Manzoni        Universitat Politécnica de València, Spain
Internal Examiner Dr. Richard Conway          University of Limerick, Ireland
Abstract

The variety of Bluetooth applications has increased greatly in recent times. The energy consumption in such applications has emerged as an important problem. The research presented in this dissertation investigates the low power operations of current Bluetooth Basic Rate/Enhance Data Rate (BR/EDR) technology and explores several novel power saving optimization approaches for applications. In order to manage power consumption in these applications, certain features in Bluetooth BR/EDR are provided to allow low-power operations, e.g. using various operation modes and packet handling processes. As such, the research focuses on Bluetooth BR/EDR technology and improving power consumption by a packet transmission efficiency protocol and optimizing design of new operation modes.

Firstly, a Packet Reassembly and Segmentation Protocol (P-RASP) in the Bluetooth baseband is proposed to operate during the idle/sleep interval duration in Bluetooth controllers. The protocol will re-assemble small host controller interface (HCI) data packets in the transmit buffers to a larger one, so that the BT link manager can assemble a larger baseband packet type with full payload.

Secondly, the research proposes a new strategy for reducing power consumption by improving the polling operation. The new approach uses a set of three different polling intervals in the Bluetooth BR/EDR controllers, whereby the controllers can choose the intervals and link state transfers from active to idle adaptively based on a common algorithm. The simulation results show this approach has very low average end-to-end packet delay and is easier and more flexible in setting the parameters than Sniff mode. Given the common algorithm or state-transition rules, a system model was established based on the Hidden Markov Model (HMM). The analysis shows the HMM can be a common model to analyze state-transition issues and be used to design and develop more efficient low power modes for Bluetooth in the future. The corresponding HMM utilization can be applied for the established system model.

Finally, the design employs a M/G(M/M)/1/N queueing model and proposes a cross-layer approach to transmit data with the Hold mode in a low rate Bluetooth-based Medical Body Area Network.

By using the proposed approaches and protocols above, the power consumption in Bluetooth can be significantly reduced. The results of this research may aid research and development teams to model, analyze and design a new operation mode to optimize power saving and improve efficiency for special Bluetooth applications.
Acknowledgements

First and foremost I would like to offer a special thanks to my supervisor, Dr. John Nelson for his guidance, knowledge and patience throughout the course of this research.

To my eternal gratitude, my parents and my wife Jing Nie, for all their love and support.

Thanks to Mr. Gordon Lessells, my best friend in Ireland, for giving me so much personal help.

I would like to thank Dr. Pepijn VandeVen, for encouraging me to do an independent research.

Thanks to the Irish Research Council for Science, Engineering & Technology (IRCSET) and the Wireless Access Research Centre in University of Limerick, for the scholarships.

To all my friends in the University of Limerick, I am infected your passion and enjoy the life with you.

A special thanks to my friends in Limerick bridge club, for kindly sharing their experience of life to me.

Finally, my sincerest gratitude to everyone I know here, I am happy with you and your friendship support and camaraderie during my time in UL.
Dedicated to my family and all my friends
# Table of Contents

Declaration iii
Abstract iv
Acknowledgements v
Dedication vi
Table of Contents vi
List of Figures x
List of Tables xiii
Nomenclature xiv

1 Introduction 1

1.1 Wireless Personal Area Networks & Bluetooth 1

1.1.1 Typically wireless technologies in PAN 2

1.1.2 Bluetooth review 3

1.2 Research context 7

1.3 Research objectives 8

1.4 Literature review 9

1.4.1 Related wireless technologies review 9

1.4.2 Relevant research in Bluetooth review 10

1.4.3 Research methods used for this research 12

1.5 Summary of research contributions 13

1.5.1 The novelty of the research 13

1.5.2 Publications arising from this work 14

1.6 Structure of this thesis 14

2 The analysis of current Bluetooth low power operations 17

2.1 Current low power operations in Bluetooth 18

2.1.1 Polling system 18

2.1.2 General operation of three low power modes 19
2.1.3 Sni mode ................................................................. 20
2.1.4 Hold mode ............................................................... 21
2.1.5 Park state ................................................................. 22
2.2 Comparison of current low power operations ......................... 22
2.3 Current issues of low power operations ................................ 23
2.4 Power consumption analysis in Sniff mode .......................... 25
  2.4.1 Sniff mode operations ................................................ 25
  2.4.2 Average power expression in Sniff mode ......................... 27
  2.4.3 Power consumption calculation and analysis ..................... 30
2.5 Summary ................................................................... 34

3 P-RASP: An Efficient Packet Reassembly and Segmentation Protocol for Bluetooth Low Data Rate Applications 36
  3.1 Related work ............................................................. 37
    3.1.1 Issue descriptions ................................................... 37
    3.1.2 Data delivery processes over the BR/EDR Controller ........ 38
    3.1.3 Analysis of solution frameworks ................................ 40
  3.2 P-RASP functionality ................................................... 41
  3.3 P-RASP operation procedures and formulas ........................ 42
    3.3.1 Negotiation by LMP or ATT and GATT ......................... 42
    3.3.2 Re-assembled packet’s identifier ................................ 43
    3.3.3 Packets Reassembly procedures ................................ 43
    3.3.4 Packet Reassembly algorithms .................................. 45
    3.3.5 Re-assembled packet segmentation algorithms ............... 46
  3.4 Performance analysis .................................................... 47
  3.5 Simulation evaluation .................................................... 48
    3.5.1 Appropriate scenarios .............................................. 48
    3.5.2 Analysis of active time ............................................. 49
    3.5.3 Set packet types and idle/sleep interval to improve P-RASP efficiency ........................................ 50
    3.5.4 Comparison of DM and DH packets by DSEIR .................. 53
    3.5.5 P-RASP in piconet with QoS option ............................. 54
  3.6 Summary and further discussion ..................................... 56

4 An Adaptive State-Transition Approach for Power Saving 58
  4.1 Adaptive state-transition approach .................................. 59
    4.1.1 The kernel idea of reduced polling operation ................. 59
    4.1.2 State machine diagram ............................................. 60
    4.1.3 Proposed approach’s state-transition rules .................... 61
  4.2 Performance analysis ................................................... 62
    4.2.1 Parameter setting .................................................... 62
    4.2.2 Context analysis ..................................................... 63
    4.2.3 Comparison with existing polling system and sniff mode .. 64
  4.3 Simulation evaluation ................................................... 66
5 The proposed state-transition approach’s system modelling and model analysis
5.1 System modelling ................................................. 77
5.2 The model analysis and parameter estimation ...................... 80
  5.2.1 The current polling system modelling ....................... 80
  5.2.2 The model parameter estimation ............................ 81
5.3 The model utilization .............................................. 83
  5.3.1 State-transition path estimation ............................ 83
  5.3.2 Average power calculation ................................. 84
  5.3.3 New state-transition rules design ......................... 85
5.4 Summary and further discussion .................................. 87

6 M/G/1/N Queueing Model and Hold mode for Cross-layer Approach 89
6.1 Related analysis .................................................. 90
  6.1.1 Characteristics and Requirements Analysis of BT-MBAN .... 90
  6.1.2 Bluetooth Hold mode review ................................ 91
  6.1.3 Using Hold mode repeatedly and the comparison with Sniff mode ............................................. 92
6.2 New design to use Hold mode in BT-MBAN ......................... 93
6.3 M/G(M/M)/1/N Queueing Model in Bluetooth ...................... 94
6.4 The operation analysis ............................................ 96
  6.4.1 Master’s operation ......................................... 96
  6.4.2 Slave’s operation .......................................... 97
6.5 Performance analysis .............................................. 97
  6.5.1 The parameters analysis by Queueing model ......... 97
  6.5.2 Optimization of Parameters by Queueing model ........ 100
6.6 Summary and further discussion .................................. 102

7 Conclusions & Recommendations 104
7.1 Research work summaries ....................................... 104
  7.1.1 Research work in Chapter 2 ................................ 104
  7.1.2 Research work in Chapter 3 ................................ 105
  7.1.3 Research work in Chapter 4 and 5 ......................... 106
  7.1.4 Research work in Chapter 6 ................................ 107
7.2 Discussion .......................................................... 107
  7.2.1 Advantage of research contributions ....................... 108
  7.2.2 Disadvantage of research contributions ............... 110
  7.2.3 Recommendations for future work ......................... 110

Bibliography 112
A  An observation sequence generation with specified probability  I

B  The simulation code of the proposed state-transition approach in Chapter 4  III

C  Publications  VII
List of Figures

1.1 Standards IEEE 802.15 for WPAN technologies. ..................................... 2
1.2 Bluetooth stack architecture. ................................................................. 5
1.3 Bluetooth piconet and scatternet’s topology. ......................................... 6
1.4 An overview of the structure of the dissertation. .................................. 15

2.1 The default polling system in Bluetooth when there is not active traffic to transmit. ............................................................. 18
2.2 The negotiation of low power modes between the hosts and the controllers. ........................................................................... 19
2.3 The place of attribute protocol and profile in the Bluetooth stack. ...... 20
2.4 Sniff mode’s operations in Bluetooth. .................................................... 21
2.5 Bluetooth packets captured by Frontline Bluetooth analyzer. .......... 24
2.6 Negotiation for Sniff mode. [1] ............................................................... 25
2.7 Sniff mode request by LMP and HCI commands [1] ......................... 26
2.8 The slave’s operations in Sniff mode. .................................................... 26
2.9 The slave’s timing operation recovery in sniff anchor point .......... 27
2.10 An example of Bluetooth operation in Sniff mode ....................... 30
2.11 Average power by using average current of Bluetooth chip in Sniff mode, $T_{sniff} = [1.25ms, 40ms]$, when no traffic. ....................... 31
2.12 Average power by using average current of Bluetooth chip in Sniff mode, $T_{sniff} = [40ms, 1.28s]$, when no traffic. ......................... 32
2.13 Average power by using average current of Bluetooth chip in Sniff mode, $T_{sniff} = [1.28s, 40s]$, when no traffic. ......................... 33

3.1 Packets from the applications are cached into queue buffer, while the peer Bluetooth node is absent from the link for idle/sleep interval duration. ................................................................. 37
3.2 L2CAP B-frame and I-frames. ............................................................... 39
3.3 L2CAP PDU transparent exchange processes in a device with a BR/EDR Controller and HCI. .................................................. 39
3.4 The structure of two solutions ............................................. 41
3.5 Request for supported P-RASP feature by link manager. .......... 43
3.6 P-RASP payload re-assembled process in baseband. ............... 45
3.7 Active slots per anchor point for each case in Scenario 1. ........... 50
3.8 Accumulated active slots in Bluetooth link during transmission time for each case in Scenario 2. ........................................ 51
3.9 The number of active slots at anchor point in Scenario 2 with different preferred packet types. ........................................ 51
3.10 The comparison of different sniff intervals and preferred packet types for Scenario 1 and 3 with P-RASP ................................. 53
3.11 Data slot efficiency improvement ratio (DSEIR) in Bluetooth. .... 54
3.12 Comparison of the maximum throughput and the actual active time for a slave in the piconet between the P-RASP and normal transmission. .................................................. 55

4.1 Peer controllers run a common algorithm to reduce polling operations. 59
4.2 The proposed state-transition approach’s state machine diagram. 60
4.3 Proposed state-transition approach’s operations when there is not data traffic to transmit over an active link. .......................... 62
4.4 The proposed state-transition approach’s reinforcement learning system architecture. .................................................. 64
4.5 The slave operations in sniff mode. ........................................ 65
4.6 An example observation sequence of a Bluetooth wireless channel simulation scenario. .................................................. 67
4.7 The average power comparison of three approaches to different traffic conditions in the Bluetooth wireless channel. ............... 69
4.8 Average end-to-end packet delay comparison of three approaches in different traffic conditions in Bluetooth wireless channel. ........ 71
4.9 Number of occupied slots of each state of three approaches in different traffic conditions in Bluetooth wireless channel. ........... 72
4.10 Different small idle interval of proposed approaches in different traffic conditions in Bluetooth wireless channel. ................... 74
5.1 The state-transition in the BT controllers with the probabilities \( a_{ij} \) and \( b_{jk} \) based on the HMM. .................................................. 79

5.2 The state-transition of the current polling system is based on a HMM with the probabilities “1” and “0” ................................................. 81

6.1 Hold request \([1]\) ......................................................................... 91
6.2 Master forces slave into Hold mode. .............................................. 92
6.3 M/G(M/M)/1/N Queueing model in Bluetooth. ............................ 94
6.4 Hold mode based approach to transmit data when stable in a BT-MBAN. ................................................................. 97
6.5 The new design’s system active time \( t_a \), based on different BT-MBAN system utilization \( \rho \) and Bluetooth transmit data rate \( \mu_a \), when set \( \lambda = 9.6 \) kbps, \( N = 4 \) kbit, \( t_h = \frac{N}{\lambda} \). .............................................. 98
6.6 The power consumption saving utilization \( \rho_{\text{saving}} \), based on different BT-MBAN system utilization \( \rho \) and Bluetooth transmit data rate \( \mu_a \), when set \( \lambda = 9.6 \) kbps. ............................................................. 99
6.7 An example calculation of whole power consumption \( F \) in a BT-MBAN, based on different Hold time value \( t_h \) at the specific scenario. 101
List of Tables

1.1 Bluetooth technology history. ............................................. 4

2.1 The comparison of the polling system and the three low power modes. 23
2.2 Bluetooth chip’s average current consumption on ACL links with
    Sniff mode. ................................................................. 31

3.1 ACL packets type and its payload on BR/EDR controllers. ........ 38
3.2 New and original logical link LLID field contents. ............... 43

4.1 The code for an observation sequence of simulation scenario. ..... 67

5.1 The state transition probability matrix $A$ and corresponding states. 79
5.2 The observation probability matrix $B$ and corresponding states. . 79
Nomenclature

Acronyms

A2MP  AMP Manger Protocol
ACL   Asynchronous Connection-oriented
ACM   Association for Computing Machinery
AMP   Alternate MAC/PHY
ASB   Active Slave Broadcast
ATT   Attribute Protocol
BR/EDR Basic Rate/Enhanced Data Rate
BT    Bluetooth
BT-MBAN Bluetooth- based Medical Body Area Network
CBR   Constant Bit Rate
CLK   Master Clock
CLKN  Native Clock
FSM   Finite State Machine
GATT  Generic Attribute Profile
HCI   Host Controller Interface
HMM   Hidden Markov Model
L2CAP Logical Link Control and Adaptation Protocol
LE  Low Energy
LMP  Link Manager Protocol
LT_ADDR  Logical Transport Address
P-RASP  Packet Reassembly and Segmentation Protocol
PALs  Protocol Adaptation Layers
PDUs  Protocol Data Units
PSB  Parked Slave Broadcast
RT  Recover Timing
SAR  Segmentation and Reassembly
SDU  Service Data Units
SIG  Special Interest Group
SSR  Sniff Sub-rating
Tcl  Tool Command Language
TDD  Time Division Duplex
UWB  Ultra-Wide Band
VBR  Variable Bit Rate
WLAN  Wireless Local Area Network
WPAN  Wireless Personal Area Network
WSN  Wireless Sensor Network
Chapter 1

Introduction

In this dissertation, the author investigates the research design, completes modelling and model analysis, and evaluates several new power saving optimization approaches in the Bluetooth [1] Wireless Personal Area Networks (WPANs).

As an introduction to the research, this chapter is structured as follows. Section 1.1 reviews Wireless Personal Area Networks and Bluetooth. Section 1.2 and Section 1.3 present the research context and issue, and research objectives. Section 1.4 presents the literature review and Section 1.5 is the summary of research contributions. Finally, Section 1.6 presents the structure of this thesis.

1.1 Wireless Personal Area Networks & Bluetooth

Wireless technologies are of great interest in recent times and here is a short summary of various WPANs and Bluetooth technologies.

A WPAN [2, 3], can be described as a network for interconnecting devices centered around an individual person’s workspace - in which the connections are wireless. A typically WPAN mostly uses unregulated bands and uses technology that permits communication within about 10 meters - in other words, a very short range. Because of the extremely short communication distances, extensive location management for a WPAN is not needed. Therefore, the applications and features for WPANs are being developed rapidly in different areas, for instance, wireless sensor network (WSN), medical body area network (MBAN), mobile Ad Hoc networks (MANETs).

While providing these applications and features, a WPAN has to achieve two main goals: broad market applicability and device interoperability. It is important
1.1. Wireless Personal Area Networks & Bluetooth

that the WPAN specifications address the devices that require wireless connectivity in a way that is both easy to implement and affordable, resulting in standards such as the IEEE 802.15. Fig. 1.1 shows the standards IEEE 802.15 for WPAN technologies [2].

Fig. 1.1: Standards IEEE 802.15 for WPAN technologies.

1.1.1 Typically wireless technologies in PAN

The technology for WPANs is in its infancy and is undergoing rapid development. Currently, most popular wireless technologies used for WPANs are Bluetooth, UWB(Ultra-wide band) and ZigBee.

- **Bluetooth**: Bluetooth is the very popular WPAN technology for connecting mobile devices, portable devices and wireless peripherals. Bluetooth was designed to support voice and data traffic simultaneity and offers raw data rates of up to 3 Mbps and ranges of up to 100 m, with lower power consumption than Wi-Fi (a main Wireless Local Area Network (WLAN) technology, based on IEEE 802.11). Bluetooth has already been widely deployed in hundreds of millions of devices and numerous Bluetooth applications have been or are being developed. Currently, the Bluetooth standard is developed by Bluetooth Special Interest Group (SIG).

- **UWB**: Ultra-Wide Band (UWB) radios emit low-power, high-bandwidth pulses that deliver data rates comparable to wired Ethernet. Its high data rates and relatively low power consumption make it ideal for replacing short wired links, like those found among PC peripherals and in home theaters. Unfortunately, the IEEE standardization of UWB (IEEE 802.15.3a) has failed, resulting in two incompatible standards: DS-UWB, advocated by the
1.1.1. Wireless Personal Area Networks & Bluetooth

UWB Forum; and MB-OFDM, advocated by the WiMedia Alliance. UWB is still in its first stages of deployment, making it difficult to predict its future success.

- **ZigBee**: Based on the standard IEEE 802.15.4, ZigBee goes even further than Bluetooth in exchanging speed for power. IEEE 802.15.4, the physical layer and medium access control of ZigBee, offers data rates of up to 250 kbps, and can easily support links with a very low duty cycle. Hence, it is suitable for deployment in battery-powered devices that must survive for up to years between charges. IEEE 802.15.4 has already found wide acceptance in the sensor network community, but like UWB, it is difficult to predict its future in other markets.

- **Others**: IEEE 802.15.5 is chartered to determine the necessary mechanisms that must be present in the PHY and MAC layers of WPANs to enable mesh networking. IEEE 802.15.7 is the standardization of Visible Light Communications (VLC). To implement special wireless applications for special targets, e.g. lower power consumption, some proprietary protocols in WPAN are developed but there is a lack of broad implementations.

1.1.2 Bluetooth review

As a popular short-range wireless communications system in PAN, Bluetooth is an industry specification for short-range RF-based connectivity for portable personal devices. The word “Bluetooth” was taken from the 10th century Danish King Harald Blatand who united dissonant Danish tribes into a single kingdom. The implication is that Bluetooth does the same with communications protocols, uniting them into one universal standard [4].

In this subsection we first review the history of Bluetooth technologies, then present the overview of Bluetooth technologies and finally summarise the future outlook of Bluetooth.

A. The history of Bluetooth technologies

The history of Bluetooth technology is shown at Table 1.1.
1.1. *Wireless Personal Area Networks & Bluetooth*

Table 1.1: Bluetooth technology history.

<table>
<thead>
<tr>
<th>years</th>
<th>Events on Bluetooth</th>
</tr>
</thead>
<tbody>
<tr>
<td>1994</td>
<td>Ericsson study a wireless technology to link mobile phones and accessories.</td>
</tr>
<tr>
<td>1999</td>
<td>Bluetooth Specification 1.0A and Bluetooth SIG promoter group expanded: 3Com, Lucent, Microsoft and Motorola.</td>
</tr>
<tr>
<td>2001</td>
<td>First retail products released, Specification 1.1 (JSR-82 Java for Bluetooth API based on Spec v1.1).</td>
</tr>
<tr>
<td>2003</td>
<td>Core Specification changes from V1.1 to V1.2, and is significantly restructured for better consistency and readability.</td>
</tr>
<tr>
<td>2004</td>
<td>IEEE 802.15.1 Study Group was closed and Bluetooth SIG continues the study of Bluetooth specification.</td>
</tr>
<tr>
<td>2004</td>
<td>Core Specification changes from V1.2 to V2.0 + EDR, introduces Enhanced Data Rate (EDR).</td>
</tr>
<tr>
<td>2007</td>
<td>Core Specification changes from V2.0 + EDR to V2.1 + EDR, and several new features are introduced.</td>
</tr>
<tr>
<td>2009</td>
<td>Core Specification changes from V2.1 + EDR to V3.0 + HS, and Alternate MAC/PHYs (AMP) are introduced as the main new feature.</td>
</tr>
<tr>
<td>2010</td>
<td>Core Specification changes from V3.0 + HS to V4.0, and Bluetooth Low Energy (BLE) is introduced as the main new feature.</td>
</tr>
</tbody>
</table>

B. Overview of Bluetooth technology

The Bluetooth technology operates in the unlicensed 2.4 GHz ISM (Industrial Scientific Medical) band and uses the 79 MHz band (2402 ~ 2480) by dividing it into 79 channels on the Basic Rate/Enhanced Data Rate (BR/EDR) Controller or 40 channels on the recently specified Low Energy (LE) Controller. The Bluetooth system employs a frequency hop transceiver to combat interference and fading. For full duplex transmission, a Time Division Duplex (TDD) scheme is used. One slot of transmitted / received data time is fixed to 625 μs and the Bluetooth BR/EDR Controller can select 1, 3 or 5 slots to transmit / receive data with different packet types and flexible packet payload. A Bluetooth system offers synchronous and asynchronous connections with data rates of 721.2 kbps for Basic Rate and up to 2.1 Mbps for Enhanced Data Rate.

The Bluetooth core system consists of a Host and one or more Controllers. A Host is defined as all of the layers below the usage profiles and above the Host Controller Interface (HCI). A Controller is defined as all of the layers below the HCI. As the Bluetooth protocol stack shows in Fig. 1.2, the baseband of Bluetooth
1.1. *Wireless Personal Area Networks & Bluetooth*

includes a BR/EDR/LE Controller and an optional Alternate MAC/PHY (AMP) Controller.

The Link Manager (LM) controls how the Bluetooth networks, known as piconets and scatternets are established and maintained by the Link Control commands. The Host Controller Interface (HCI), with UART, USB, SD and three wire UART transport options, provides a uniform command method of accessing controller capabilities.

The Bluetooth host includes a series of protocols to implement various applications. The Logical Link Control and Adaptation Protocol (L2CAP) provides connection oriented and connectionless data services to upper layer protocols with protocol multiplexing capability and segmentation and reassembly operation.

The basic topology of Bluetooth-based WSNs is called a piconet, which is the fundamental form of communication in the Bluetooth Basic Rate / Enhanced Data Rate (BR/EDR) wireless technology. A piconet consists of two or more devices that occupy the same BR/EDR physical channel. The master BT node can only communicate with one active slave of BT node at a given time slot and have up to seven active slave nodes. A group of piconets in which connections exist between different piconets is called a scatternet in which a Bluetooth device that
is a member of two or more piconets is involved. The piconet and scatternet’s topology are shown in Fig. 1.3.

![Bluetooth piconet and scatternet's topology.](image)

The Version 3.0+HS Specification of the Bluetooth system, which was released on 21st April, 2009, added an Alternate MAC/PHY (AMP) Controller including AMP Protocol Adaptation Layers (PALs) and AMP Manager Protocol (A2MP). The AMP provides the operation for incorporating an extraneous device, for example an 802.11 device compliant with the 2007 edition of the IEEE 802.11 Standard.

The Bluetooth Low Energy controller was first introduced by the Bluetooth Core Specification Version 4.0 in December 2009 to cater for very low rate sensor network applications, which included the related Low Energy physical and link layers, as well as enhancements to the HCI for Low Energy, Low Energy direct test mode and AES encryption. This version has been adopted as of June 30, 2010.

C. Bluetooth future outlook

In the future, the trend of technology development will be toward technology fusion rather than creation of new technology. Bluetooth is suitable for this.

Bluetooth core specification V3.0 supports theoretical data transfer speeds of up to 24 Mbit/s, though not over the Bluetooth link itself. Instead, the Bluetooth link is used for negotiation and establishment, and the high data rate traffic is carried over a collocated 802.11 link. Two technologies had been anticipated for AMP: 802.11 and UWB, but UWB is missing from the specification. The AMP greatly increases the re-use of Bluetooth upper-layer protocols.

Bluetooth core specification V4.0 introduces the Bluetooth LE controller, which focuses on the applications of low rate data wireless sensor networks (WSN) and
1.2. Research context

contributes a very low power module. The Bluetooth LE has too low a data rate for most WPAN applications. However, the dual module Bluetooth device in the centre network can access both Bluetooth BR/EDR and LE controllers. This means that the Bluetooth LE enriches the access mode of WPAN for a restrained single module Bluetooth device, e.g. a sport watch, which may only have a Bluetooth LE controller.

In addition, Bluetooth BR/EDR technology is proven and hundreds of millions of Bluetooth BR/EDR devices have been deployed. Further improvement of existing devices is also an important area of technology development.

1.2 Research context

More and more mobile and portable devices have a Bluetooth wireless module. These devices have a large battery to supply for up to a week or more. Where Bluetooth module is the main data transmission module between devices, its energy consumption has emerged as an important problem.

The research has investigated the low power operations of current Bluetooth BR/EDR technology and the emerging Bluetooth LE technology. The studies show that although the Bluetooth BR/EDR technology provides many low power operations and the appropriate use of the standardized low power modes can lead to significant power saving. There are difficulties in using current low power operations related to the setting of parameters and it is difficult to compare performance between the response time and energy efficiency. Therefore, to date, most Bluetooth applications have not adopted the current low power operations just to reduce the response time. Bluetooth LE technology is not the complete power saving solution either: it is necessary to integrate a new wireless chipset to support the low data rate applications. This means that Bluetooth LE isn’t backwards compatible and has a very low data rate that will limit its applicability. Hence, Bluetooth LE will not supersede the Bluetooth BR/EDR.

The focus of this research is on the proven Bluetooth BR/EDR controllers which have been widely used to support a broad range of applications requiring low to high rate data throughput. In order to improve power consumption in these applications, variety of optimal designs of new operation modes and a packet transmission efficiency protocol will be provided in this dissertation.
1.3. Research objectives

The main research objectives, direction and approaches were to investigate low-power operations in Bluetooth BR/EDR technology, with a view to designing new optimization approaches to power saving for use with Bluetooth BR/EDR technology. The contents and objectives of the research in detail were as follows:

- To study the state-of-the-art Bluetooth and relevant Bluetooth application requirements.

- To investigate in particular the existing and emerging sensors used in Medical body area networks; to analyze the current wireless technologies and solutions in sensor networks; to define a number of typical scenarios for the targeted network, for example, activity sensors and vital signs sensors.

- To investigate the Bluetooth technology power saving strategies and analyze the current low-power operations, e.g. Sniff mode and sniff sub-rating; Hold mode and Park state.

- To investigate strategies for cross layer optimization in wireless sensor networks; to perform Cross-layer optimization for Bluetooth for power saving; and to analyze and propose new features or protocols in Bluetooth.

- To re-design and implement current low-power modes for a low rate data specified application, e.g. medical body area networks.

- To design a new protocol for packet Segmentation and Reassembly (SAR) and to optimize the packet transmission efficiency and improve the performance of power saving in Bluetooth.

- To research on the adaptive low power operation scheduling scheme based on novel algorithms with corresponding modelling analysis, for instance, the Hidden Markov Model (HMM) and the queueing model for Bluetooth wireless networks.

- To evaluate the performance of the new research designs, in conjunction with the current low-power operations in Bluetooth for typical WSNs applications.
1.4 Literature review

In order to achieve the above research objectives, the review of relevant literature, which are presented in the bibliography section of this dissertation, have been studied throughout the course of this research. A number of the primary references of significant interest to the research is presented in this section.

1.4.1 Related wireless technologies review

Wireless communications has changed dramatically over the past few years. Recently, the main changes are the ability to exploit the transmission capacities of radio channels and the applications that rely on wireless transmission [5]. Based on the physical properties of radio and the regulation of wireless, the wireless technologies, which include Satellite systems, Cellular system, wireless LAN and PAN, and many others, can be divided by the use of frequency bands and these bands are assigned by government agencies. A single type of wireless system can offer many types of services, e.g. mobile telephones offer paging services and Bluetooth technology offers a variety of voice and data services. By Harte and et al. the book [5] covers the basics of wireless technologies.

The research is more concerned about the short-range wireless technologies. The WPAN and Bluetooth technologies [1, 2] are briefly summaried in Section 1.1. The Wireless Local Area Network (WLAN) [6] systems operate in the 2.4 GHz (IEEE 802.11b and 802.11g) and 5.7 GHz (IEEE 802.11a) frequency bands, and can be used in both indoor and outdoor environments and has longer range than Wireless PAN. The IrDA (Infrared Data Association) [7] and RFID (Radio-Frequency Identification Device) [8] technologies have been studied for comparative examples. These technologies above have been reviewed in numerous books [5, 8, 9], or specifications [6, 7].

The rising UWB and ZIGBEE technologies have been studied in specifications [2] and other papers [10, 11]. Some wireless technologies are designed for specific applications. For example, a wireless sensor network (WSN) has important applications such as remote environmental monitoring and target tracking. These sensors are small, with limited processing and computing resources, and they are inexpensive compared to traditional sensors [12]. The WSN technologies have been reviewed in papers [12, 13].
1.4. Literature review

1.4.2 Relevant research in Bluetooth review

Bluetooth-related research spans a very broad array of electronic and computer engineering, and network applications. Bluetooth have been studied largely from the specification [1], which includes the architecture, the required technical details, the core system package of Host & Controllers, and Host Controller interface (HCI). Many articles and books [14] introduce the basics of developing applications using the Bluetooth. The subsection shall discuss relevant research of the power saving topic in Bluetooth.

A. The low data rate applications in Bluetooth

With the increase in applications [15, 16, 17, 18, 19, 20, 21, 22] of Bluetooth-based WSNs, many related problems of Bluetooth-based WSNs are exposed such as [12]. These WSN applications [15, 16, 17, 18, 19], which are Asynchronous Connection-oriented (ACL) data packet based [20], usually have a low data rate (typically 3 ~ 300 kbps) and can accept longer packet delay (often in the level of seconds). A Bluetooth system offers synchronous and asynchronous connections with data rates of 721.2 kbps for Basic Rate (BR) and up to 2.178 Mbps for Enhanced Data Rate (EDR). In [21], the Bluetooth transceiver is configured as a wireless serial port and utilizes the UART transmit and receive functions of the respective PIC and Bluetooth devices. The serial transmission rate is 115.2 kbps, which is much less than the capacity of Bluetooth. In [22], the designed fall sensor, which reports recorded fall events and alerts, uses Bluetooth to configure device and log data. The real-time fall events are sent to a SD memory card first due to Bluetooth’s poor transmission efficiency for low data rates. Thus, the power saving issue about low data rate applications in Bluetooth is: how to reduce the throughput or the duty cycle on a Bluetooth link and still cater for the required rate of applications.

B. The polling scheduling scheme in Bluetooth

The problem of high power consumption of the Bluetooth polling system has been highlighted for many of the broad range of ACL applications. Therefore, many studies [23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33] propose various new polling schemes to improve power consumption in Bluetooth. For example, in [23], the authors propose a traffic-aware algorithm to deal with the polling and scheduling issue within a piconet (i.e. a network of Bluetooth devices, controlled by a master with
1.4. Literature review

One or more slaves); in [24], a dynamic scheduling scheme with the implementation of direct communication between any two slaves within a piconet is proposed; the authors of [25] focus on the problem of designing efficient and fair, yet simple to implement, polling schemes that would achieve best performance (i.e. the lowest end-to-end packet delay) under TCP traffic; papers [26, 27, 28, 29] address traffic aware and cross-layer optimization of polling schemes; energy-efficient error control with packet selection and scheduling schemes in interference environments are proposed in [30, 31, 32]; while in [33], the master sends NULL packets with sufficient frequency to maintain synchronization of the slave, which doesn’t have to respond and thereby can save some power.

C. Low power modes

In the core specification, the Bluetooth Special Interest Group (SIG) provides three low power operation modes for the Basic Rate/Enhanced Data Rate (BR/EDR) controllers to manage power consumption and defines the Bluetooth low energy (LE) controller to cater for low data rate sensor network applications.

The studies [34, 35] show that the appropriate use of the standardized low power modes can lead to significant power saving. The authors of [36, 37, 38, 39, 40, 41] proposed various low power mode scheduling schemes for power saving. The white paper [36] is a technology report by Bluetooth SIG about how to use sniff/sniff sub-rating modes. In [37], to change the sniff parameters, a learning function is used to approximate the distribution of the incoming traffic at a master-slave pair. The authors of [38] proposes to use a two dimensional matrix for the representation of sniff attempts and managing the low-power sniff mode in Bluetooth. The paper [39] proposes repeated use of Hold mode to deliver data for power saving and the paper [40] proposes a new interpiconet scheduling algorithm based on the Hold mode to reduce the average interpiconet packet delay while increasing the utilization of a bridge. The paper [41] proposed a self-referencing structure: Blue-Park, where each node parks all the other nodes as a master. The packets are exchanged between nodes with dynamic park/unpark procedures, which results in dynamic assignment of master/slave roles.

D. Bluetooth low energy technology

The Bluetooth low energy (BLE) is an emerging wireless, low-power technology, which is unique for the device’s ability to keep going for months or even years.
on hardly any battery power using the new Bluetooth LE controller. From the specification [1], the Bluetooth LE system divides the 2.4 GHz ISM band into 40 RF channels. These RF channels have center frequencies $2402 + k \times 2 \text{ MHz}$, where $k = 0, ..., 39$. The RF channels 0, 12 and 39 are advertising channels with indexes 37, 38, 39 respectively. Other RF channels are data channels with indexes 0 to 36. The Bluetooth LE controller can enter the connection state either from the initiating state or the advertising state in relatively fast time, in the order of milliseconds. During the connection state, the LE controller doesn’t use POLL-NULL packets but uses 1 bit More Data (MD) at the head of a data channel packet to indicate that the device has more data to send or not. The connection event shall continue till the MD of both controllers is equal to “0”. Then the connection closes.

The paper [42] presents an analytical model for the maximum throughput of BLE as a function of bit error rate and a BLE parameter called connInterval. The paper [43] gives a design for a medical device, which is a BLE implantable glucose monitoring system. Many companies have also announced their support of Bluetooth LE, and devices have recently become available.

1.4.3 Research methods used for this research

The most common research methods are: literature searches, system modelling for the issues, analysis and simulation tools.

The most complete technologies information of this research is from the Bluetooth specification [1]. There are many web sites that provide retrieval of articles and allow access to the full text. The main related web sites of this research are the IEEE web site, the ScienceDirect web site, the Elsevier web site and the association for computing machinery (ACM) web site.

In order to analyze the research issues, some mathematical models are used. Hidden Markov model (HMM) and relevant algorithms have been studied [44, 45, 46, 47, 48]. A Hidden Markov Model (HMM) is a double stochastic process with one underlying process that isn’t observable but may be estimated through a set of processes that produce a sequence of observations [46]. In this research, the HMM will be used to analyze the state-transition issue in Bluetooth.

Queueing theory is the mathematical study of waiting lines, or queues and has been studied [49, 50, 51]. In [51], the performance of a single Bluetooth piconet is analyzed using the theory of M/G/1/N queues with vacations. In this research,
1.5. Summary of research contributions

The queueing model will be used to analyze the system utilization and system parameters optimization while Hold mode is repeatedly used in Bluetooth.

Simulation tools for Bluetooth have been studied, which includes Blueware [52] and UCBT [53] under ns-2, and MATLAB [54]. The simulation of Bluetooth scenarios often uses Blueware [52] or UCBT (University of Cincinnati - Bluetooth) [53], which are NS-2 based Bluetooth network modules. These two modules implement a near complete Bluetooth (specification 1.1 or 2.0) stack, and include support of the Bluetooth Baseband, LMP, and host layers such as L2CAP. MATLAB is a software package used primarily in the field of engineering for signal processing, numerical data analysis, modeling, programming, simulation, and computer graphic visualization [54].

In addition, the Bluetooth analyzer [55] has been studied and used for Bluetooth protocol analysis and Bluetooth packet sniffing.

1.5 Summary of research contributions

The main emphasis of the research project was to investigate the design, modelling and model analysis, and evaluation of several new power saving optimization approaches in the Bluetooth Wireless Personal Area Networks (WPANs). We can summarise the research contributions throughout the course of this research as follows.

1.5.1 The novelty of the research

The novel contributions of this research project can be described as follows:

In order to enhance the packet transmission efficiency in Bluetooth, a Packet Reassembly and Segmentation Protocol (P-RASP) is proposed to re-assemble small host controller interface (HCI) data packets in the transmit buffers to a larger one, so that the Bluetooth link manager can assemble a larger baseband packet type (e.g. 3-DH5) with full payload. This research work is presented in Chapter 3.

This research proposes a new strategy for reducing power consumption by improving the polling operation. The new approach uses a set of three different polling intervals in the Bluetooth BR/EDR controllers, whereby the controllers can choose the intervals and link state transfers from active to idle adaptively based on a common algorithm. The simulation results show this approach has very low average end-to-end packet delay and is easier and more flexible in setting the
parameters than Sniff mode. Given the common algorithm or state-transition rules, a system model was established based on the Hidden Markov Model (HMM). The analysis shows the HMM can be a common model to analyze state-transition issues and be used to design and develop more efficient low power mode in Bluetooth in the future. The corresponding HMM utilization can be applied for the established system model. This research work is presented in Chapters 4 and 5.

This research proposes using Hold mode to transmit data in a Bluetooth-based Medical Body Area Network (BT-MBAN), which has lower aperiodic data rates. The employment of an M/G(M/M)/1/N queueing model and Hold mode in Bluetooth can improve power efficiency significantly for data transmission in many MBAN scenarios. This research work is presented in Chapter 6.

1.5.2 Publications arising from this work

A number of technical papers have been published during the course of this research work. In total four conference papers and one journal paper have been in process, and one journal paper has been submitted. Appendix C shows a more detailed list of the publications.

1.6 Structure of this thesis

This section outlines briefly the content of each chapter in the dissertation. An outline is also provided in Fig. 1.4.

Chapter 1: A review of the available literature is provided in this chapter. The research topics and issues of this dissertation are power saving approaches for use with Bluetooth.

Chapter 2: The Bluetooth SIG provides a variety of low-power operations. A detailed analysis of the polling system and the three low modes are presented in this chapter. The study will compare these low-power operations and make recommendations for their uses.

Chapter 3: A Packet Reassembly and Segmentation protocol (P-RASP) in the Bluetooth baseband is proposed to re-assemble small host controller interface (HCI) data packets in the queue buffer to a larger one, so that the BT link manager can assemble a larger baseband packet type (e.g. 3-DH5) with full payload. The P-RASP core functionality, the corresponding operation and procedure details and performance evaluation within appropriate scenarios are studied. Experimental
1.6. Structure of this thesis

Fig. 1.4: An overview of the structure of the dissertation.
results show that the P-RASP can reduce the active time of Bluetooth link and offer efficient packet transmission.

**Chapter 4**: In this chapter, a new approach recommends a reduction of polling operations by setting three different polling intervals, which can be small, medium and large in the controllers. Each controller runs a common algorithm to choose among the three polling intervals and adaptively transfers link state from active to idle while maintaining system synchronization. The analysis reveals that this approach can be implemented autonomously on the controllers and significantly improve Bluetooth power efficiency by reducing the polling operations.

**Chapter 5**: In order to further study an improved parameter settings or state-transition rules and algorithms for power saving, this chapter establishes a mathematical model of the proposed approach by the Hidden Markov Model (HMM). The established system model can be used to estimate state-transition paths on the controllers and to calculate the average power of the proposed approach for a generic or general example. The model also can be used to design new state-transition rules for special applications. Actually, the research reveals the HMM can be a generic model to analyze the state-transition issues and be used to design and develop more efficient low power modes for Bluetooth in the future.

**Chapter 6**: This chapter proposes using a M/G/1/N queueing model and the Hold mode to transmit data in low rate networks, such as a Bluetooth-based Medical Body Area Network (BT-MBAN). The specific operations of the master and the slaves are given. The employment of an M/G(M/M)/1/N queueing model and Hold mode in Bluetooth can improve power efficiency significantly for data transmission in many MBAN scenarios.

**Chapter 7**: The chapter summaries the results and contributions of the research.
Chapter 2

The analysis of current Bluetooth low power operations

Bluetooth provides various low power operations to manage power consumption. At the microscopic level, the operation of packet handling and slot occupancy must be minimized. The basic idea is to reduce information exchange between the Bluetooth devices, and allow the transmitter and receiver to return to sleep if possible. At the macroscopic level, the basic idea which realizes low power is to adopt low power operations that reduces the duty cycle of the Bluetooth devices.

In this chapter, the current Bluetooth low power operations are reviewed. For an active Bluetooth link, the polling system is adopted by default. When there is not traffic to transmit, the packet handling and slot occupancy can be reduced by the polling system. The designers also can use certain low power modes—Sniff, Hold and Park in Bluetooth specification to save power. Through using the polling system or the three low power modes, the power consumption of Bluetooth-based applications can be reduced, but the channel re-synchronization is needed and it may have other problems, e.g. packet delay. Thus, the proper use of low power operations and their current issues shall be discussed.

The chapter is structured as follows. Section 2.1 explains the polling system in Bluetooth and gives a short summary of the three low power modes. Section 2.2 gives a comparison of low power operations and in Section 2.3 the current issues of low power operations are analyzed. Section 2.4 details the analysis of power consumption in Sniff mode, which includes the detailed operation, the expression of average power and the evaluation of the Bluetooth’s power. Finally, Section 2.5 presents the conclusion.
2.1 Current low power operations in Bluetooth

The current low power operations in Bluetooth include the polling system and the three low power modes—sniff, hold and park, all of which are optional. They are explained below.

2.1.1 Polling system

Bluetooth operates on a master/slave concept and adopts time division multiplexing (TDM), where a slave cannot transmit until a master poll it. To maintain system clock synchronization when there is not active traffic from the higher layers, such as the L2CAP layer and above, the master sends a POLL packet to the slave who shall send back a NULL immediately to acknowledge; then after ten sequential POLL-NUL pairs, the unnecessary consecutive polling operations shall be reduced and the master sends a POLL to the slave using the selected poll interval, which has a default value of 40 slots (every 25 ms, since each slot is 625 $\mu$s). Therefore, the POLL-NUL packets are adaptively reduced and the Bluetooth controller, in the default case, can have up to 38 idle slots after each POLL-NUL when there is not active traffic to transmit.

Fig. 2.1 shows the current polling system in Bluetooth. When Bluetooth establishes an Asynchronous Connection-oriented (ACL) link with no active traffic from the upper layer, the link with the default polling system will typically have 40 poll operations per second after 10 sequential POLL-NUL operations as required by the polling system specification. In addition, the slave is not expected to sleep during the idle period as the master can send data or POLL packets anytime.

![Fig. 2.1: The default polling system in Bluetooth when there is not active traffic to transmit.](image-url)
2.1. Current low power operations in Bluetooth

2.1.2 General operation of three low power modes

These three low power modes must be explicitly activated in an active Bluetooth link.

A. The basic parameters negotiation by HCI and LMP commands

First-off, each of the low power mode parameters needs negotiation. As Fig. 2.2 shows, the basic operation of the three low power modes, where the low power mode and corresponding parameters are negotiated between hosts and implemented by the BR/EDR controllers, using Host Controller Interface (HCI) and Link Manager Protocol (LMP) commands.

Fig. 2.2: The negotiation of low power modes between the hosts and the controllers.

B. The advanced parameters negotiation by ATT and GATT

The Bluetooth SIG has developed new operating modes and protocols to address Bluetooth for low data rates and asynchronous data. The Attribute protocol (ATT) [56] provides a method to communicate small amounts of data over a fixed L2CAP channel. The attribute protocol has notification and indication capabilities that provide an efficient way of sending attribute values to a client without the need for them to be understood. The Attribute protocol is also used by devices to determine the services and capabilities of other devices.

The Generic Attribute Profile (GATT) defines a service framework using the Attribute Protocol. This framework defines procedures and formats of services and their characteristics. The definition includes discovering, reading, writing, notifying and indicating characteristics, as well as configuring the broadcast characteristics. Fig. 2.3 shows the place of Attribute protocol and profile in Bluetooth stack.
2.1. Current low power operations in Bluetooth

![Layer Diagram]

Fig. 2.3: The place of attribute protocol and profile in the Bluetooth stack.

Thus, the ATT and GATT can be a common method to complete applications info exchange between Bluetooth devices, which improve the compatibility and interoperability of connected devices such as medical sensors.

C. The general operations and issues

When the current traffic conditions have met the set low power parameters from the upper layers, the controller can transfer to an idle or sleep state, which is based on the specified low power state. (In idle state, the master can poll the slave or the slave can receive packets from a master, but typically they can the act of nothing or no work. In the sleep state, the master can not poll a slave and the slave can not receive for more aggressive power saving.) For each mode, the active duty cycle of Bluetooth devices is reduced, by eliminating the unnecessary POLL-NULL packets used for channel synchronization but at the expense of re-synchronizing after the sleep period and with packet delay if there is traffic during the sleep period.

The master can poll the slave if both of them in the idle state;

2.1.3 Sniff mode

The polling system with the poll interval 40 slots is the default setting in an active Bluetooth link. However, as an alternative, Sniff mode is the recommended low power mode for low data rate applications by the Bluetooth SIG.

In Sniff mode, the master and the slave agree periodic anchor points where they will communicate. Consequently, the Bluetooth devices negotiate a sniff interval ($T_{\text{sniff}}$) and a sniff offset ($D_{\text{sniff}}$) in the ACL logical link. The master shall only start
2.1. Current low power operations in Bluetooth

a transmission to the slave in the specified sniff attempt slots and sniff timeout slots, and the slave may return to sleep in the remaining slots of the $T_{sni}$ period. The slave device keeps its logical transport address (LT_ADDR) in the idle or sleep state. The sniff sub-rating (SSR) allows both the master and slave to increase the time between sniff anchor points, which can further reduce power consumed by link management in Sniff mode.

Fig. 2.4 shows Sniff mode’s operations in Bluetooth. After running the sniff attempt slots and sniff timeout slots, the slave will go to idle/sleep for power saving and the duty cycle of the Bluetooth devices is reduced. The more detailed operations of Sniff mode will be discussed in subsection 2.4.1.

![Fig. 2.4: Sniff mode’s operations in Bluetooth.](image)

Sniff mode is designed for periodic or aperiodic low data rate applications, e.g. low data rate sensor networks. If sniff attempt and sniff timeout parameters are properly set, Sniff mode can also run during high data rate applications, e.g. large file transfer.

### 2.1.4 Hold mode

In Hold mode, the master slave link becomes inactive and doesn’t support ACL packets on the piconet’s channel. A timer is initialized with the timeout value $holdT0$ (hold time) and the slave device can enter a sleep state. When the timer expires, the slave wakes up, synchronizes to the traffic on the channel and waits for further master transmissions. Hold mode applies to unreserved slots on the physical link. When in this mode, the physical link is only active during slots that are reserved for the operation of the synchronous link types SCO and eSCO. All
2.2. Comparison of current low power operations

asynchronous links are inactive. Hence, the ACL logical link in Bluetooth can be paused by suspending data traffic during the hold time, and reduce the polling operation between Bluetooth devices which means power saving. During Hold mode, the slave device keeps its LT_ADDR.

Hold mode is designed to “run only once” for each invocation, which means both devices exit Hold when complete and the previous mode, e.g. active or sniff mode, is restored. Therefore, Hold mode was not considered in the specification as a low power operation and it is designed as a network scheduling method for the piconet or scatternet in Bluetooth. The holdTO parameter often is set to the level of seconds. However, Hold mode can be used repeatedly by sending command for power saving, which is addressed in Chapter 6.

2.1.5 Park state

From the specification [1], a slave in the park state doesn’t need to participate on the piconet’s channel, but still needs to remain synchronized to the channel. The slave shall give up its LT_ADDR and shall receive two new addresses to be used in the park state.

The parked slave wakes up at regular intervals to listen to the channel in order to re-synchronize and to check for broadcast messages. To support the synchronization and channel access of the parked slaves, the master supports a beacon train described in the next section. The beacon structure is communicated to the slave when it is parked. When the beacon structure changes, the parked slaves are updated through broadcast messages. The master shall maintain separate non-overlapping park beacon structures for each hop sequence. The beacon structures shall not overlap in either their beacon slots or access windows.

The park state is used to connect more than seven slaves to a single master in the piconet. However, park state is mentioned here for completeness, but its use is discouraged due to the complexity of its parameters and the feature is expected to be deprecated and removed from the core specification in the near future [36].

2.2 Comparison of current low power operations

In power saving, the polling system and the three low power modes are designed for different scenarios. Different scenarios and different network requirements may
2.3. *Current issues of low power operations*

result in different operations. The comparison of low power operations is as shown in Table 2.1.

Table 2.1: The comparison of the polling system and the three low power modes.

<table>
<thead>
<tr>
<th>Feature / capability</th>
<th>Polling system</th>
<th>Sniff mode</th>
<th>Hold mode</th>
<th>Park mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active state</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No, in Park state</td>
</tr>
<tr>
<td>LT_ADDR</td>
<td>Keep</td>
<td>Keep</td>
<td>Keep</td>
<td>Give up</td>
</tr>
<tr>
<td>Transmit</td>
<td>Any time</td>
<td>At sniff attempt and timeout slots</td>
<td>Quit Hold mode</td>
<td>Quit Park State</td>
</tr>
<tr>
<td>re-synchronization</td>
<td>Immediate</td>
<td>Fast</td>
<td>Fast</td>
<td>Slow</td>
</tr>
<tr>
<td>Packet delay</td>
<td>Short and up to 38 slots as default</td>
<td>$T_{\text{sniff}}$, up to 40.9 s</td>
<td>$\text{holdTO}$ value, typically seconds, up to 40.9 s</td>
<td>The longest, no guarantee</td>
</tr>
<tr>
<td>Power saving</td>
<td>Good</td>
<td>Good</td>
<td>Very good</td>
<td>The best</td>
</tr>
<tr>
<td>Appropriate scenarios</td>
<td>Default, used for all applications</td>
<td>Low data rate applications, e.g. WSNs</td>
<td>Network scheduling</td>
<td>More than 7 nodes</td>
</tr>
</tbody>
</table>

2.3 **Current issues of low power operations**

The operations and comparison of the polling system and the low power modes have been briefly described in the last two sections. This section will discuss the issues arising they are used.

The Frontline Bluetooth analyzer [55] was adopted to capture Bluetooth packets in a link. It can be seen in Fig. 2.5 that there are many POLL-NUL packet pairs between the intervals of arriving DM1 data packets, when the controllers have only control packets to exchange or have low rate data packets arriving from the upper layer. The reason is that the packets have “best effort” delivery while system synchronization is maintained by POLL-NUL packets in the default Bluetooth polling system.
2.3. Current issues of low power operations

<table>
<thead>
<tr>
<th>B. Freq</th>
<th>Chn</th>
<th>R.</th>
<th>CLK</th>
<th>Packet Type</th>
<th>Type</th>
<th>LLID</th>
<th>FLOW</th>
<th>L2CAP Flow</th>
<th>Len</th>
<th>Delta</th>
</tr>
</thead>
<tbody>
<tr>
<td>40,961</td>
<td>12</td>
<td>M.</td>
<td>0x700174</td>
<td>OK</td>
<td>POLL</td>
<td>N/A</td>
<td>N/A</td>
<td>0</td>
<td>8210</td>
<td></td>
</tr>
<tr>
<td>40,962</td>
<td>12</td>
<td>S.</td>
<td>0x700176</td>
<td>OK</td>
<td>NULL</td>
<td>N/A</td>
<td>N/A</td>
<td>0</td>
<td>8290</td>
<td></td>
</tr>
<tr>
<td>40,963</td>
<td>12</td>
<td>M.</td>
<td>0x700178</td>
<td>OK</td>
<td>POLL</td>
<td>N/A</td>
<td>N/A</td>
<td>0</td>
<td>8210</td>
<td></td>
</tr>
<tr>
<td>40,964</td>
<td>12</td>
<td>S.</td>
<td>0x700179</td>
<td>OK</td>
<td>DMT</td>
<td>L2CAP</td>
<td>Go</td>
<td>17</td>
<td>8210</td>
<td></td>
</tr>
<tr>
<td>40,965</td>
<td>10</td>
<td>M.</td>
<td>0x70017a</td>
<td>OK</td>
<td>POLL</td>
<td>N/A</td>
<td>N/A</td>
<td>0</td>
<td>8290</td>
<td></td>
</tr>
<tr>
<td>40,966</td>
<td>10</td>
<td>S.</td>
<td>0x70017b</td>
<td>OK</td>
<td>NULL</td>
<td>N/A</td>
<td>N/A</td>
<td>0</td>
<td>8200</td>
<td></td>
</tr>
<tr>
<td>40,967</td>
<td>10</td>
<td>M.</td>
<td>0x700180</td>
<td>OK</td>
<td>POLL</td>
<td>N/A</td>
<td>N/A</td>
<td>0</td>
<td>8220</td>
<td></td>
</tr>
<tr>
<td>40,968</td>
<td>20</td>
<td>S.</td>
<td>0x700182</td>
<td>OK</td>
<td>NULL</td>
<td>N/A</td>
<td>N/A</td>
<td>0</td>
<td>8290</td>
<td></td>
</tr>
<tr>
<td>40,969</td>
<td>40</td>
<td>M.</td>
<td>0x700184</td>
<td>OK</td>
<td>POLL</td>
<td>N/A</td>
<td>N/A</td>
<td>0</td>
<td>8220</td>
<td></td>
</tr>
<tr>
<td>40,970</td>
<td>40</td>
<td>S.</td>
<td>0x700186</td>
<td>OK</td>
<td>NULL</td>
<td>N/A</td>
<td>N/A</td>
<td>0</td>
<td>8290</td>
<td></td>
</tr>
<tr>
<td>40,971</td>
<td>12</td>
<td>M.</td>
<td>0x700188</td>
<td>OK</td>
<td>POLL</td>
<td>N/A</td>
<td>N/A</td>
<td>0</td>
<td>8220</td>
<td></td>
</tr>
<tr>
<td>40,972</td>
<td>12</td>
<td>S.</td>
<td>0x70018a</td>
<td>OK</td>
<td>NULL</td>
<td>N/A</td>
<td>N/A</td>
<td>0</td>
<td>8290</td>
<td></td>
</tr>
<tr>
<td>40,973</td>
<td>12</td>
<td>M.</td>
<td>0x70018c</td>
<td>OK</td>
<td>DMT</td>
<td>L2CAP</td>
<td>Go</td>
<td>1016</td>
<td>8220</td>
<td></td>
</tr>
<tr>
<td>40,974</td>
<td>20</td>
<td>S.</td>
<td>0x70018e</td>
<td>OK</td>
<td>NULL</td>
<td>N/A</td>
<td>N/A</td>
<td>0</td>
<td>3120</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 2.5: Bluetooth packets captured by Frontline Bluetooth analyzer.

Therefore, the default polling system need handle several sequential POLL-NULL pairs, which have been shown in Fig. 2.5. However, the default polling system cannot reduce the polling adaptively while the value of polling interval is set, even if the application can tolerate longer packet delay. The three low power modes, especially Sniff mode, are used to deal with this situation by cross-layer negotiation of parameters.

The cross-layer negotiation between the hosts and the controllers often cannot meet the requirement of applications in view of the difficulty of setting parameters. In Sniff mode, the master will only transmit to a slave at regular times specified by the sniff interval. The sniff interval differs per application and is dependent on the required data latency and throughput rate. It is necessary for an application to do lots of interoperability testing and try different combinations of sniff parameter values to see which works best when considering both power consumption and responsiveness. Therefore, the designers often set different parameters for different applications and do lots of duplicate test work.

Hold mode is designed as a single shot to allow a device to complete another activity such as discovering other devices and it enables the peer device(s) to enter low power. A slave in PARK state doesn’t need to participate on the piconet’s channel but needs to remain synchronized to the channel. Park mode requires that the controllers negotiate and maintain various park related parameters which has resulted in very limited implementations. Like Sniff mode, these two low power modes are initiated and controlled by applications through a set of HCI commands, however there is little guidance for application usage and in particular Park mode is rarely used and is frequently not made available to device users.
2.4 Power consumption analysis in Sniff mode

Most research work focuses on Sniff mode and it is widely recommended for low power operation. In this section, the Bluetooth power consumption in Sniff mode within different sniff intervals will be discussed in detail.

2.4.1 Sniff mode operations

Currently, Sniff mode is the most common and flexible method for reducing Bluetooth’s power consumption. The operations are as follows.

A. LMP and HCI Commands Operations

To enter Sniff mode, both the master and slave can start a negotiation through the Link Manager Protocol (LMP) messages, commonly referred to as LMP protocol data units (PDUs). The process is initiated by sending an LMP_sniff_req PDU containing a set of parameters, which includes timing control flags, $D_{\text{sniff}}$, $T_{\text{sniff}}$, $N_{\text{sniff,attempt}}$, $N_{\text{sniff,timeout}}$. The receiving LM shall then decide whether to 1) reject the attempt by sending an LMP_not_accepted PDU; 2) suggest different parameters by replying with an LMP_sniff_req PDU; or 3) accept the request. The negotiation is shown at Fig. 2.6.

![Fig. 2.6: Negotiation for Sniff mode.](image)

The HCI_Sniff_Mode command is used to alter the behavior of the LM and have it place the ACL baseband connection associated with the specified Connection Handle into Sniff mode. The HCI_Exit_Sniff_Mode command is used to end Sniff mode for a Connection Handle, which is currently in Sniff mode.

The HCI_Sniff_Mode command has five command parameters, which are Connection_Handle, Sniff_Max_Interval, Sniff_Min_Interval, Sniff_Amount and Sniff_Timeout. Note the HCI_Sniff_Mode parameters include a min and max sniff
2.4. Power consumption analysis in Sniff mode

interval, allowing the LMP a degree of flexibility in selecting the $T_{sni}$ period. An example of a Sniff mode request which is accepted is shown in Fig. 2.7.

![Fig. 2.7: Sniff mode request by LMP and HCI commands [1] ](image)

**B. Transmitter and Receiver Operations**

When a slave enters Sniff mode, it need not listen at every receive slot (Rx) and its receiver may go to sleep until the next anchor point. There are two key parameters in Sniff mode: $N_{sni\_attempt}$ and $N_{sni\_timeout}$. These parameters specify the number of baseband receive slots for sniff attempt and timeout, respectively. The slave continues to listen from the anchor point for the specified number of sniff attempts, and if it has received a packet addressed to it, then it may continue to listen for more packets until the specified timeout. The slave’s receiver prepares to receive packets from the master at the scheduled sniff receive slots and to do some operations based on the content of received packets or pending data for transmission. Fig. 2.8 gives an overview of the operations of the slave transmitter and receiver.

![Fig. 2.8: The slave’s operations in Sniff mode. ](image)
2.4. Power consumption analysis in Sniff mode

From Fig. 2.8, we observe that the slave has a recover timing (RT) period at the sniff anchor point. The reason is the slave runs using its native clock (CLKN) in Sniff mode and loses synchronization while sleeping. The master clock (CLK) is used for all timing and scheduling activities in the piconet. The slave maintains its own approximation to the master clock, but due to timing jitter and time drift between the respective clocks, it must continuously resynchronize. Hence, an uncertainty window is defined around the exact receive timing. The slave shall not recover the master timing until it receives a packet including the piconet access code from the master at the sniff attempt slots. The slave’s recover timing operation is shown in Fig. 2.9 where the recover timing window is centred around the slave’s estimation of the sniff anchor point time.

![Diagram of Sniff anchor point](image)

Fig. 2.9: The slave’s timing operation recovery in sniff anchor point

2.4.2 Average power expression in Sniff mode

The Bluetooth’s power consumption in Sniff mode will be scaled by the average power during sniff interval. We analyze the different states when the Bluetooth device is in Sniff mode and calculate the time spent in each state \( t_i \) \[57\]. Therefore the average power can be expressed by

\[
P_{\text{avg}} = \frac{\sum P_i \times t_i}{\sum t_i} \tag{2.1}
\]

Where \( P_i \) represents the power consumption in state \( i \).

A. Slave’s Average Power

In Sniff mode, the CLKN may be driven by a low power oscillator with worst case timing accuracy (specified in the standard as drift = +/- 250 ppm and jitter =
2.4. Power consumption analysis in Sniff mode

10 μs), therefore, the slave’s Rx recover timing period (t_{RT}) should be considered. The drift time parameter (t_{drift}) and jitter time parameter (t_{jitter}) will incur power consumption due to re-synchronization. The RT power in Sniff mode P_{RT}(S) can be the averaged power in a sniff interval, which is as shown in:

\[
P_{RT}(S) = P_{drift}(S) + P_{jitter}(S) = \frac{(2 \times drift \times T_{sniff} + 2 \times t_{jitter}) \times i_{Rx}}{T_{sniff}} \times V \tag{2.2}
\]

S is the slave. T_{sniff} is Sniff mode interval; i_{Rx} is the current of a device in slave role when receiving; V is the voltage for the specific Bluetooth chip. To simplify the analysis, we set \(N_{sniff,timeout}=0\). The t_{slot}’s length is 625 μs. Although it is possible to go to idle after the packet has been received, we will assume that it remains receiving for the full slot. Therefore, from the description of sniff operations in subsection 2.4.1, the slave’s Rx consumed average power \(P_{Rx}(S)\) in Sniff mode is

\[
P_{Rx}(S) = \frac{(N_{sniff,attempt} \times t_{slot}) \times i_{Rx}}{T_{sniff}} \times V \tag{2.3}
\]

If the master has no traffic to the slave during the sniff intervals, it is recommended that single slot packets are transmitted by the master during the slave re-synchronization. Therefore the slave should be sent a POLL or NULL packet which includes the piconet access code so that the slave shall keep synchronized to the channel. The slave’s transmitter will acknowledge, at the corresponding transmission slot (Tx) e.g. using a NULL packet if no command or data needs to be sent. The slave in the other Tx opportunities typically won’t send any packet if it has nothing to send. The slave’s Tx consumed average power \(P_{Tx}(S)\) in Sniff mode is

\[
P_{Tx}(S) = \frac{i_{Tx} \times t_{slot} + ((N_{sniff,attempt} - 1) \times t_{slot}) \times i_{T_x, idle}}{T_{sniff}} \times V \tag{2.4}
\]

\(i_{Tx}\) is the current of a device in slave role when transmitting; \(i_{T_x, idle}\) is the current of a device in slave role when not transmitting and idle.

After \(N_{sniff,attempt}\), the slave enters sleep until the next sniff anchor point, which means the lowest power consumption while the slave is in connection state. The sleep consumed average power in Sniff mode is

\[
t_{sleep} = T_{sniff} - (2 \times drift \times T_{sniff} + 2 \times t_{jitter}) - N_{sniff,attempt} \times 2 \times t_{slot}
\]

\[
P_{sleep}(S) = \left(\frac{t_{sleep} \times i_{sleep}}{T_{sniff}}\right) \times V \tag{2.5}
\]
2.4. Power consumption analysis in Sniff mode

The average power in Sniff mode for the basic scenario considered is the sum of every operation’s average power, which is given by:

\[
P_{\text{avg}}(S) = [P_{\text{RT}}(S) + P_{\text{Rx}}(S)] + P_{\text{Tx}}(S) + P_{\text{Sleep}}(S)
\]  

(2.6)

B. Master’s Average Power

The master has three typical operations in Sniff mode on the ACL link. The first is when the master sends a POLL packet to the slave and receives a NULL packet in return and continues to do so in sniff intervals, which means the effective sniff interval \( T_{\text{sniff}} = 1.25\text{ms}(2\text{ slots}) \) and the master’s slots are always involved \( (N_{\text{poll}} = 2\text{ slots}) \) on the ACL links. This is the worst case of power consumption that the master device always run in active and not in idle, even if there is no traffic. The master’s average power \( P_{\text{avg, on}}(M) \) is as follows.

\[
P_{\text{avg, on}}(M) = \frac{i_{\text{Tx}} \times \frac{1}{2} T_{\text{sniff}} + i_{\text{Rx}} \times \frac{1}{2} T_{\text{sniff}}}{T_{\text{sniff}}} \times V = \frac{1}{2}(i_{\text{Tx}} + i_{\text{Rx}}) \times V
\]  

(2.7)

The second is when the master works as in normal ACL with no data traffic to be sent in sniff intervals, which means the master only sends POLL or NULL packets within the poll internal \( N_{\text{poll}} \), which can be the default value 40 slots. This is the same as the polling system’s operation with the same \( N_{\text{poll}} \) while stable. The master’s average power \( P_{\text{avg, poll}}(M) \) is as follows.

\[
P_{\text{avg, poll}}(M) = \left( \frac{(i_{\text{Tx}} + i_{\text{Rx}}) + i_{\text{idle}} \times (N_{\text{poll}} - 2)}{N_{\text{poll}}} \right) \times V
\]  

(2.8)

The third typical operation is when the master is on the ACL link with or without data transfer, which means the master slots are used for other logical transport traffic or for POLL and NULL packets to the slave which might be in active or idle/sleep in Sniff mode. The approximate power consumption of this work state is between \( P_{\text{avg, on}}(M) \) and \( P_{\text{avg, poll}}(M) \).

Fig. 2.10 illustrates an example of Bluetooth operation during sniff for both master and slave devices, while there is no traffic. Idle denotes that the receiver attempts to receive but after the short receive window, realizes that there is no packet being transmitted and changes to the idle state.
2.4. Power consumption analysis in Sniff mode

2.4.3 Power consumption calculation and analysis

A. Calculate Average Power

The power consumption of a Bluetooth module is the sum of that of the Bluetooth chip and other parts of the module e.g. microprocessor or wired communication devices. If a Bluetooth chip works in Sniff mode, other parts of the module e.g. three-line UARTs could be set to sleep to save power. We only consider the power saving of a Bluetooth chip in Sniff mode.

The traffic between Bluetooth devices on the ACL link is also variable and different applications with different Sniff mode parameters are often required. To simplify the analysis, we only consider that there is no traffic between devices, which means that the devices only send POLL or NULL packets on the $N_{\text{sni, attempt}}$ slots and therefore the $N_{\text{sni, timeout}}$ will be set to 0.

Different Bluetooth chips have different Tx or Rx current and voltage parameters. A simplifying assumption is that voltage parameter is a normalized constant (e.g. $V=1$) and use the current’s unit (microampere) to scale Bluetooth’s power.

From [58] [59], we obtain indications about the range of the current parameters in Bluetooth chip. Therefore we can estimate the values of the parameters, which are representative of real device values. We set the parameters as follows: $i_{T_x} = 22 \text{ mA}$, $i_{R_x} = 18 \text{ mA}$, $i_{T_x, idle} = i_{idle} = 4 \text{ mA}$, $i_{sleep} = 40 \mu\text{A}$, $N_{\text{sni, attempt}} = 1$ and $N_{\text{sni, timeout}} = 0$. Considering the time drift parameter is variable and is not always specified at the max value. We set its average value $\text{drift}_{\text{avg}} = 0.5 \times \text{drift}_{\text{max}} = 125\text{ppm}$.

Therefore, we can make use of equations (2.2) - (2.8) to calculate average power by using average current. The result is in Table 2.2.
2.4. Power consumption analysis in Sniff mode

Table 2.2: Bluetooth chip’s average current consumption on ACL links with Sniff mode.

<table>
<thead>
<tr>
<th>Connection types</th>
<th>Operations Mode (on ACL links)</th>
<th>Average</th>
<th>Unit</th>
<th>( \frac{P(S)}{P(M)} ) (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>master device</td>
<td>( N_{\text{poll}} = 2 ) slots (The worst case)</td>
<td>20.0</td>
<td>mA</td>
<td>-</td>
</tr>
<tr>
<td>master device</td>
<td>( N_{\text{poll}} = 40 ) slots (Use default)</td>
<td>4.80</td>
<td>mA</td>
<td>-</td>
</tr>
<tr>
<td>slave device</td>
<td>( T_{\text{sni}} = 1.25 ) ms (CLK(_2))</td>
<td>20.29</td>
<td>mA</td>
<td>101%</td>
</tr>
<tr>
<td>slave device</td>
<td>( T_{\text{sni}} = 40 ) ms (CLK(_7))</td>
<td>0.677</td>
<td>mA</td>
<td>3.4%</td>
</tr>
<tr>
<td>slave device</td>
<td>( T_{\text{sni}} = 1.28 ) s (CLK(_{12}))</td>
<td>0.0642</td>
<td>mA</td>
<td>0.32%</td>
</tr>
<tr>
<td>slave device</td>
<td>( T_{\text{sni}} = 40.9 ) s (CLK(_{17}))</td>
<td>0.0454</td>
<td>mA</td>
<td>0.23%</td>
</tr>
</tbody>
</table>

B. Power Consumption Analysis

From Table 2.2, we observe that if \( T_{\text{sni}} = 1.25 \) ms (2 slots), the Bluetooth chip of the slave will not save power in Sniff mode and the slave’s power will be consumed more than the master’s power. The reason is the slots are always involved (\( N_{\text{poll}} = 2 \) slots) on the ACL links. The more power are consumed by the recover timing operation (channel re-synchronization) in the slave, which have been shown in Fig. 2.9. If \( T_{\text{sni}} = 40 \) ms, the chip of the slave will save 96.6% power consumption, compared to the master’s average power consumed using \( N_{\text{poll}} = 2 \) slots on the ACL links. Fig. 2.11 compares and analyzes the master and slave’s power calculation results for different sniff intervals.

![Fig. 2.11: Average power by using average current of Bluetooth chip in Sniff mode, \( T_{\text{sni}} = [1.25\text{ms},40\text{ms}] \), when no traffic.](image)

The “*” and “+” lines in the figure shows the master device’s average power by using average current with \( N_{\text{poll}} = 2 \) slots or 40 slots (25 ms), respectively. Mostly the master device’s average power is between “*” and “+” line, and the \( T_{\text{sni}} \) axes is insignificant. The continuous “-” line in the figure shows the slave device’s average power within \( T_{\text{sni}} = [1.25\text{ms},40\text{ms}] \). The graph indicates that the larger value of \( T_{\text{sni}} \) will provide more power saving, while there is no traffic between devices.
The slave’s power calculation results for Sniff mode with different sniff intervals [40ms,1.28s] are simulated in Fig. 2.12.

Fig. 2.12: Average power by using average current of Bluetooth chip in Sniff mode, $T_{\text{sni}} = [40\text{ms},1.28\text{s}]$, when no traffic.

The continuous “-.” line in the figure shows the slave device’s average power by using average current within $T_{\text{sni}} = [40\text{ms},1.28\text{s}]$.

The graph indicates that the larger value of $T_{\text{sni}}$ will provide more power saving, while there is no traffic between devices.

In Fig. 2.12, if $T_{\text{sni}} = 40.9\text{ s}$ and $N_{\text{sni attempt}} = 1$, the slave role will save more power in theory. In reality, devices should not adopt the longest $T_{\text{sni}}$ to save power even when the application allows it. There are a few things that must be considered besides power saving.

First of all, we must consider the clock time drift of slave and the channel conditions. If in perfect synchronization then the slave device always receives the master to slave transmission at the sniff anchor point. However, the slave CLKN time drift could result in $2 \times drift \times T_{\text{sni}}$ 10-20 ms drift in the case $T_{\text{sni}} = 40.9\text{ s}$; whereby the slaves might lose timing and shall require channel re-synchronization before it may send information. The channel re-synchronization of the slave consumes power dependent on the length of a single search window and the duration of the search.

Simultaneously, the slave may lose traffic from the master depending on channel conditions. In the worse case channel conditions it is possible that an LMP supervision timeout will be reached (max 40.9 s if enabled) causing the ACL connection to be lost. Thus, it is recommended that the parameter $N_{\text{sni attempt}} \times t_{\text{slot}}$ is larger than $2 \times drift_{\text{max}} \times T_{\text{sni}}$ in Sniff mode to improve the
2.4. Power consumption analysis in Sniff mode

The probability that the slave receives a packet scheduling from master during the sniff attempt windows. The slave will consume more power than the above calculation indicates in the longer sniff intervals.

Secondly, for most Bluetooth applications it is important that the device has a rapid response time and minimizes latency when communicating to the other device. The longer the sniff interval, the longer waiting time when the device needs to transmit data. During Sniff mode, some applications may require to switch between idle or sleep to active mode with high frequency. Consequently, if the power saving of Bluetooth is only the Bluetooth chip in Sniff mode, it is meaningless to improve the power saving from 96.6% to 99.9% by using the longest sniff interval 40.9 s.

In addition, sniff sub-rating (SSR) provides a mechanism to increase the time between sniff anchor points. Even if SSR increases the response time in the normal case the advantage of SSR is that when a packet is missed at a SSR anchor point the slave can listen and recover synchronization at the next sniff anchor points. Therefore, using SSR can also result in more efficient power saving instead of using the longest $T_{\text{sni}}$.

Finally, there are implications when selecting the value of the $N_{\text{sni\_attempt}}$ parameter. From a power consumption perspective, Fig. 2.13 shows the impact of $N_{\text{sni\_attempt}}$ and $T_{\text{sni}}$.

![Fig. 2.13: Average power by using average current of Bluetooth chip in Sniff mode, $T_{\text{sni}} = [1.28s,40s]$, when no traffic.](image)

The continuous “-”, “+” and “*” lines in the figure shows the slave device’s average power by using average current when $N_{\text{sni\_attempt}} = 1, 5, 10$ respectively. The graph indicates that the smaller value of $N_{\text{sni\_attempt}}$ will provide more power saving, while there is no traffic between devices.
2.5 Summary

This chapter introduced the current low power operations in Bluetooth BR/EDR controller. The polling system with the poll interval 40 slots is the default setting in an active Bluetooth link. The three low power modes—Sniff, Hold, Park, are needed to be set by cross-layer negotiation between hosts and to be implemented by the BR/EDR controllers. For each mode, the active duty cycle of Bluetooth devices is reduced, by eliminating the unnecessary POLL-NULL packets used for channel synchronization but at the expense of re-synchronizing after the sleep period and increased packet delay if it has traffic during the sleep period. The polling system is default on active ACL links and can be used in all the scenarios; Sniff mode was suggested for use in low rate sensor networks; Hold mode was suggested for use in network scheduling and Park state was suggested for use in the piconet when the piconet has more than 7 slave nodes.

The current issues of low power operations was further discussed. The analysis showed the default polling system cannot handle several sequential POLL-NULL pairs; and the three low power modes often cannot meet the requirement of applications in view of the difficulty of setting parameters.

Sniff mode is currently the most common and flexible method for reducing Bluetooth’s power consumption. The expressions for average Bluetooth power consumption in Sniff mode are given through analysis of the operations of Bluetooth. The Bluetooth average power in Sniff mode with different sniff intervals are evaluated. The analysis shows, while there is no traffic between devices, the Bluetooth chip in slave mode can save 96.6% of its power consumption when in Sniff mode, based on realistic current parameters and setting $T_{\text{sniff}} = 40$ ms. The saving is in comparison to the master’s power consumed when $N_{\text{poll}} = 2$ slots on the ACL links. If the Bluetooth devices enter Sniff mode with an appropriate $T_{\text{sniff}}$, the power consumption of devices can be reduced. Generally, the Bluetooth devices should not adopt the longest $T_{\text{sniff}}$ to save power due to the channel re-synchronization and data latency issues.

Based on the analysis above, the current low power operations, especially Sniff mode, could reduce the power consumption of Bluetooth-based applications. However, in reality, the parameters of low power operations are not easy to set to cater for the requirements of the variety of applications.

Hence, there is a requirement to develop more usable approaches, through simplification without expensive modification or addition of complex features.
2.5. Summary

resultant new design of the low power operations and protocols will be explored in the following dissertation chapters.
Chapter 3

P-RASP: An Efficient Packet Reassembly and Segmentation Protocol for Bluetooth Low Data Rate Applications

Studies indicate that during data transmission of Bluetooth low data rate ACL applications, Bluetooth devices have to run with small size baseband packets (e.g. 3-DH1) and Bluetooth controllers have to do many polling operations to maintain system synchronization. The default polling system and the sniff mode in Bluetooth standard can reduce unnecessary polling operations in low rate applications, and allow Bluetooth devices to enter a low power state, which is an idle/sleep interval duration in the BR/EDR controller.

In this chapter, during the idle/sleep interval duration, a Packet Reassembly and Segmentation Protocol (P-RASP) in the Bluetooth baseband is proposed to reassemble small host controller interface (HCI) data packets in the transmit buffers to a larger one, so that the Bluetooth link manager can assemble a larger baseband packet type (e.g. 3-DH5) with full payload.

The chapter is structured as follows. Section 3.1 explains related work about the issues. Section 3.2 describes the P-RASP protocol’s core functionality and Section 3.3 describes the P-RASP’s operation procedures and formulas. The performance analysis and simulation evaluation within appropriate scenarios are detailed in Section 3.4 and Section 3.5. Finally, Section 3.6 presents the summary and further discussion of this research.
3.1 Related work

This section will discuss the related issue descriptions, technology detail and solution frameworks while a low data rate application runs at the Bluetooth BR/EDR controller.

3.1.1 Issue descriptions

Using Bluetooth BR/EDR controller to make WPANs or WSNs has many challenges within low data rate applications, for instance, 1) how to reduce Bluetooth throughput and still cater for the required rate of applications and 2) how to enhance the packet transmission efficiency.

To address the former issue, the generic operations allow the Bluetooth nodes to have an agreed idle/sleep interval duration on an active link to reduce the active time of the link. The Bluetooth Special Interest Group (SIG) provides low power modes in the baseband specification, e.g. sniff mode, to reduce the frequent polling operations. In the case of a piconet, if the master is transmitting data to a slave, the other slaves in the piconet can enter sniff mode using an agreed idle/sleep interval duration by pre-set sniff parameters.

The packets, which come from applications during the idle/sleep interval duration, shall be buffered into a queue in the transmit buffer, which is shown in Fig. 3.1.

![Fig. 3.1: Packets from the applications are cached into queue buffer, while the peer Bluetooth node is absent from the link for idle/sleep interval duration.](image)

The latter issue about “packet transmission efficiency” is caused by different packet types and payloads in Bluetooth. The Bluetooth baseband has sixteen different types of packets (ranging from NULL, POLL, FHS, DM1 to 3-DH5).
3.1. Related work

On an asynchronous connection-oriented (ACL) data link, seven packet types are defined for BR operation: DM1, DH1, DM3, DH3, DM5, DH5 and AUX1; six additional packets are defined for EDR operation: 2-DH1, 3-DH1, 2-DH3, 3-DH3, 2-DH5 and 3-DH5. The larger payload packets have higher throughput and efficiency. Table 3.1 shows the ACL packets type on the BR/EDR Controller and each of the payload ranges.

Table 3.1: ACL packets type and its payload on BR/EDR controllers.

<table>
<thead>
<tr>
<th>Packets Types</th>
<th>DM1</th>
<th>DH1</th>
<th>DM3</th>
<th>DH3</th>
<th>DM5</th>
<th>DH5</th>
<th>AUX1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Payload(bytes)</td>
<td>0-17</td>
<td>0-27</td>
<td>0-121</td>
<td>0-183</td>
<td>0-224</td>
<td>0-339</td>
<td>0-29</td>
</tr>
<tr>
<td>Packets Types</td>
<td>2-DH1</td>
<td>3-DH1</td>
<td>2-DH3</td>
<td>3-DH3</td>
<td>2-DH5</td>
<td>3-DH5</td>
<td></td>
</tr>
<tr>
<td>Payload(bytes)</td>
<td>0-54</td>
<td>0-83</td>
<td>0-367</td>
<td>0-552</td>
<td>0-679</td>
<td>0-1021</td>
<td></td>
</tr>
</tbody>
</table>

These packet types differ in payload, error coding and take up a different number of slots (either 1, 3 and 5 slots). The Link Manager (LM) in the controller assembles 1, 3 & 5 slot packet types based on the packet size from the upper layer as appropriate.

However, when we observe the transmission of low rate applications in Bluetooth, the active link always has many polling operations and typically uses 1 slot baseband packets (e.g. DM1, DH1, 2-DH1 and 3-DH1) to deliver data. There are two possible reasons: the first is that the application’s data rate is low and the upper layer protocols (e.g. BNEP, RFCOMM, OBEX, SDP) encapsulate the data to small sized service data units (SDU), so that the size of L2CAP packet data units (PDU) is also small; the other reason is the Bluetooth devices may have resource restrictions and set the Maximum Transmission Unit (MTU) of L2CAP to the minimum 48 bytes.

Before proposing a solution for these issues, the data delivery processes over the Bluetooth BR/EDR controller are next reviewed.

3.1.2 Data delivery processes over the BR/EDR Controller

When the ACL data of the upper layer applications, protocols, or profiles arrive at L2CAP, in the form of an SDU, the L2CAP will encapsulate them into one or more L2CAP PDUs based on the L2CAP MTU. The range of Bluetooth BR/EDR L2CAP MTU can be set between the minimum 48 bytes and the maximum 64K
3.1. Related work

bytes and the default value is 672 bytes. The L2CAP PDU has two kinds of packet formats in a connection-oriented channel, a Basic Information Frame (B-frame) in Basic L2CAP mode and an Information Frame (I-frame) in the retransmission/flow control/streaming modes. When the length of an SDU is bigger than the MTU in L2CAP, the L2CAP will segment and encapsulate an SDU fragment in multiple I-frames. Fig. 3.2 shows how the L2CAP encapsulates an SDU to one or more L2CAP PDUs.

![Fig. 3.2: L2CAP B-frame and I-frames.](image1)

The L2CAP PDU usually is exchanged between the host and controller by the transport mechanism of the HCI. The exchanged data format adopts HCI ACL data packets. Fragmentation operations can be done to break down L2CAP PDUs into smaller pieces for delivery from L2CAP to the controller to be carried in the payload of Bluetooth baseband packets. The Packet_Boundary_Flag (PB) in the HCI packet is used to indicate if the HCI ACL Data payload is a total L2CAP PDU or a fragment. The Bluetooth baseband packets also shall indicate it by using the Logical Link Identifier (LLID) in the baseband payload header. An LLID code of ‘10’ indicates the first segment in an L2CAP PDU and ‘01’ for a continuation segment. Fig. 3.3 shows the L2CAP PDU transparent exchange processes in a device with a BR/EDR controller and HCI. The buffered HCI ACL packets shall

![Fig. 3.3: L2CAP PDU transparent exchange processes in a device with a BR/EDR Controller and HCI.](image2)
3.1. Related work

be mapped to Bluetooth baseband packets and be delivered when the Bluetooth channel is available.

If the upper layer applications’ data rates are low and the size of SDU is relatively small, e.g. 27-81 bytes L2CAP SDU packet including the header from the upper layer, the SAR function of L2CAP needn’t work and a DH3 or 3-DH1 baseband packet can be assigned with the LLID code ‘10’. In addition, if the L2CAP MTU is set to the minimum value 48 bytes, the L2CAP PDU shall be 48 bytes or smaller and a DH3 or 2-DH1 baseband packet shall be assigned with the LLID code ‘10’. In either case, we can observe that many small size HCI ACL packets are queued in the controller buffers.

3.1.3 Analysis of solution frameworks

To maintain system synchronization while supporting low rate applications, the unnecessary polling operations often happen and can be reduced by an agreed idle/sleep interval duration in the controller. During the idle/sleep interval duration, the incoming HCI ACL data shall be queued in the buffer of the controller. If there is not enough room in the buffer, the HCL ACL data will be queued in the upper layer buffers. The buffered packets in the controller will result in small payload baseband packets (e.g. DM1, DH3, 3-DH1), which will be delivered one by one over the Bluetooth baseband channel. Therefore, Bluetooth BR/EDR controllers have to use many small size baseband packet types to deliver the corresponding small payload packets during low rate applications, which is resource inefficient.

There are two solutions which have different solution frameworks on the BR/EDR controller. One framework is that data packets can be reassembled at the application level. For example, a wearable physiologic sensor [60] stores data to a 32 MB of on-board flash memory and the data can be sent as a larger SDU when it needs to be delivered by Bluetooth. The L2CAP’s SAR function shall be invoked if necessary i.e. the large SDU will be segmented as appropriate. The other framework is that data packets can be reassembled in the Bluetooth controller. Fig. 3.4 shows the structure of two solutions.

The first solution framework is often used in health devices, for example, devices which record a person’s physical or physiological activity and often have low application data rates. The recorded data is used for subsequent analysis. Therefore, the applications in these health devices usually don’t need real-time
3.2. P-RASP functionality

Fig. 3.4: The structure of two solutions

data transmission by Bluetooth. If the devices need real-time data transmission, and need to confine the data transmission, the Bluetooth LM can suspend data traffic on the L2CAP channel, i.e. by the hold mode \([1]\) and cache the data in the application level buffers. The Bluetooth LM has to give notice to L2CAP when the channel is available or not and also give notice to the peer device \([39]\), which means cross-layer negotiation \([61]\). Thus, it is challenging and has much repetitive overhead work as pre-set parameters for each application are needed.

The second solution framework needn’t change the data transmission of applications and the application will remain unaware. The Bluetooth BR/EDR controller is required to reassemble small size packets in the baseband and send at the appropriate time. There is significant opportunity for higher power efficiency in Bluetooth BR/EDR controllers, if the small HCI data packets, e.g. carried in DM1, DH3 or 3-DH1 baseband packets, can be re-assembled to enable delivery as a larger data packet, e.g. 3-DH5, and the controller goes to sleep and eliminates the polling operations. This solution requires a new packet reassembly and segmentation protocol.

### 3.2 P-RASP functionality

The issue for low rate Bluetooth-based sensor network applications, as already shown in the previous sub-section, includes the transmission of many small size baseband packets and frequent polling operations in the classic BR/EDR controller.

In particular, there are various opportunities for energy saving during idle/sleep
3.3. P-RASP operation procedures and formulas

intervals for low rate WSN applications. Periodic sleep intervals can be used by enabling the sniff mode, or can be estimated by setting appropriate QoS parameters. The existing Bluetooth protocols don’t provide the opportunity to take full advantage of these idle/sleep intervals by adopting more efficient baseband packet payloads. Hence, a new protocol is required typically running during the idle/sleep interval duration of the Bluetooth BR/EDR controller and the result is the Packets Reassembly and Segmentation Protocol (P-RASP).

The core functionality is as follows:

- When the Bluetooth device has an idle/sleep interval duration and small payload HCI ACL data packets are buffered into the BR/EDR controller buffer, the LMP can re-assemble the small size packets selectively during the idle/sleep interval duration before delivery.

- When the Bluetooth receiver receives a re-assembled baseband packet, the LM can identify it and the packet can be segmented and restored to the original small payload HCI ACL data packets in the controller.

- The operations above shall be negotiated by LM commands and implemented in the controller.

3.3 P-RASP operation procedures and formulas

In this section, details of how P-RASP would be incorporated into the Bluetooth specification are presented.

3.3.1 Negotiation by LMP or ATT and GATT

The P-RASP can be an extended feature in the BR/EDR controllers. The Bluetooth device can negotiate it by a new LM command LMP_PRAS and use LMP_accepted or LMP_not_accepted to respond to it. Fig. 3.5 shows a request for the supported P-RASP feature by the Link Manager (LM) in the communicating Bluetooth devices.
3.3. P-RASP operation procedures and formulas

![Diagram of LM and LMP PRAS](image)

Fig. 3.5: Request for supported P-RASP feature by link manager.

As for the parameters negotiation of low power modes, the ATT and GATT, which was described in subsection 2.1.2, also can be used to negotiate the P-RASP functionality. The ATT and GATT is often used in low data rate applications and provides advanced parameters negotiation which can include more parameter options for a special application such as the P-RASP.

### 3.3.2 Re-assembled packet’s identifier

The re-assembled packet shall be encapsulated in a Bluetooth baseband packet and identified by the other Bluetooth device. The Logical Link Identifier(LLID) code ‘00’ in the baseband packet’s payload header is currently undefined in the standard. Therefore, the research recommends using LLID code ‘00’ in the payload header to indicate that a re-assembled L2CAP message is being transported. The new and original details about the contents of the LLID field are listed in Table 3.2.

Table 3.2: New and original logical link LLID field contents.

<table>
<thead>
<tr>
<th>LLID code</th>
<th>Logical Link</th>
<th>Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>ACL-U</td>
<td>A re-assembled L2CAP message. (new)</td>
</tr>
<tr>
<td>01</td>
<td>ACL-U</td>
<td>Continuation fragment of an L2CAP message</td>
</tr>
<tr>
<td>10</td>
<td>ACL-U</td>
<td>Start of an L2CAP message or no fragmentation</td>
</tr>
<tr>
<td>11</td>
<td>ACL-C</td>
<td>LMP message</td>
</tr>
</tbody>
</table>

### 3.3.3 Packets Reassembly procedures

During the sleep slots, for example of a sniff interval period, the controller shall not send (or receive) any HCI ACL data packets to (or from) the peer Bluetooth device (but may send HCI SCO packets or attend to another piconet). The HCI ACL data packets from L2CAP, being sent to the peer device, shall stay in the controller ACL
3.3. P-RASP operation procedures and formulas

buffer if there is enough room (as indicated by LM command Read_buffer_size), or stay in the upper layer buffer.

In principle, 1) the P-RASP runs at the idle/sleep interval durations of the connection state in the controller; 2) the ACL control logical link (ACL-C) packets won’t be re-assembled; 3) the P-RASP only re-assembles the HCI ACL data packets with a common user ACL link (ACL-U), which is identified by the HCI packets connection handle.

The first condition is because Bluetooth delivers packets with best effort as default, the P-RASP shall not change to introduce delay; the second is because the ACL-C packets need to be processed directly and the peer might be waiting for the response; the last is that different ACL-Us imply that the packets have a different connection handle and might have a connection to different Bluetooth nodes.

When the P-RASP is running and a new HCI ACL data packet arrives, the P-RASP Packet Reassembly operations in the LM can be described as follows:

- If the current buffered HCI ACL packet is an ACL-C packet, or is an ACL-U with a different connection handle, or there is inadequate capacity in the current baseband packet, then the arriving packet will be processed as normal.

- If the current buffered packet is ACL-U packet with a common connection handle and there is capacity in the current baseband packet, then the arriving packet shall be reassembled with the current buffered packet. First check the baseband packets LLID. If the buffered packet’s LLID code is ‘01’ or ‘10’, insert a new payload header with LLID code ‘00’, and set the appropriate flow control bit and the length. If the current packet’s LLID code is ‘00’ insert the payload of new packet after the current packet payload and update the current baseband payload header length in the baseband packet’s header.

The payload re-assembly process in the baseband is shown in Fig. 3.6.
3.3. P-RASP operation procedures and formulas

3.3.4 Packet Reassembly algorithms

The P-RASP can be defined as follows.

\[ P = \text{the packet}; \quad L = \text{the packet length}; \]
\[ i = \text{the sequence index for packets } P \text{ coming from HCI while reassembling (or to HCI while segmenting)}; \]
\[ j = \text{the sequence index for packets } P \text{ queued in the buffer}; \]
\[ k = \text{the sequence index for packets } P \text{ segmented}; \]
\[ P_{HCI}^i \text{ the HCI ACL packet}; \quad P_{BUF}^j \text{ the buffered packet}; \]
\[ L_{\text{pref}} = \text{the preferred } P \text{ length}; \]

At initialization, set \( i = 1, \quad j = 1 \) and \( P_{BUF}^1 = \text{empty}. \)

Whenever, if the \( P \) is an ACL-C packet, then the \( P \) shall be converted to a baseband packet and buffered in the buffer directly (we don’t re-assemble ACL-C packets). Thus, the operations statements in the controller is shown in formula \((3.1)\).

\[
P_{BUF}^j \leftarrow P; \quad j = j + 1; \tag{3.1}
\]

The first line in formula \((3.1)\) shows the normal operation in the controller where the packet is inserted into the buffer and will be converted to a Bluetooth baseband packet.

If the \( P_{HCI}^i \) is an ACL-U packet, and the current baseband packet \( P_{BUF}^{j-1} \) doesn’t exist or exists and is an ACL-C packet or is an ACL-U with a different connection handle than that of \( P_{HCI}^i \), then the \( P_{HCI}^i \) shall be converted to a baseband packet and queued in the buffer directly. Thus, the operations statements in the controller is the same as the formula \((3.1)\) above.

If the \( P_{HCI}^i \) is an ACL-U packet and \( P_{BUF}^{j-1} \) is also ACL-U packet with same connection handle, then the \( P_{HCI}^i \) will NOT be re-assembled to \( P_{BUF}^{j-1} \) if the length
3.3. P-RASP operation procedures and formulas

of the re-assembled packet becomes more than $L_{\text{pref}}$. Thus, if $L(P^i_{\text{HCI}})+L(P^{i-1}_{\text{BUF}}) > L_{\text{pref}}$, the operations statements in the controller is the same as the formula (3.1) above.

If the $P^i_{\text{HCI}}$ and $P^{i-1}_{\text{BUF}}$ are ACL-U packets with same connection handle and the $L(P^i_{\text{HCI}})+L(P^{i-1}_{\text{BUF}}) \leq L_{\text{pref}}$ does hold, then the $P^i_{\text{HCI}}$ shall be assembled with $P^{i-1}_{\text{BUF}}$. Thus, the operations statements in the controller is shown in formula (3.2).

$$
\begin{align*}
P^j_{\text{BUF}} &\leftarrow P^i_{\text{HCI}}; \\
P^{j-1}_{\text{BUF}} &\leftarrow P^{j-1}_{\text{BUF}} + P^j_{\text{BUF}}; \quad \% \text{Reassembly.} \\
L(P^{j-1}_{\text{BUF}}) &\leftarrow L(P^{j-1}_{\text{BUF}}) + L(P^j_{\text{BUF}}); \\
\text{LLID}(P^{j-1}_{\text{BUF}}) &\leftarrow '00'; \\
P^j_{\text{BUF}} &\leftarrow \text{empty}; \\
i &\leftarrow i + 1;
\end{align*}
$$

(3.2)

In formula (3.2), the first line is the normal operation in the controller; the second line is the reassembly operation by P-RASP; the other lines are to generate a new Bluetooth baseband header for current buffered packet $P^{i-1}_{\text{BUF}}$.

3.3.5 Re-assembled packet segmentation algorithms

When the controller receives a packet with the LLID set to code ‘00’, the LM and HCI shall segment the packet’s payload from the length byte fields in payload headers, and restore the original packets to the HCI ACL data packets. Therefore from then on the normal operation of the Bluetooth controller will be carried out transparently.

At the initialization, set $i = k = 1$ and $j = 1$. Therefore, the $P^{k=1}_{\text{BUF}}$ is the first segmented packet and the $P^{j+1}_{\text{BUF}}$ is the packet after the $P^{j}_{\text{BUF}}$ is segmented. If the LLID code ‘00’, the packet segmentation operations shall run in the controller till $L(P^{j}_{\text{BUF}}) = 0$. Thus, the operation formula in the controller is shown in formula (3.3).

$$
\begin{align*}
P^k_{\text{BUF}} &\leftarrow P^j_{\text{BUF}}; \quad \% \text{Segmentation.} \\
P^{j+1}_{\text{BUF}} &\leftarrow P^j_{\text{BUF}} - P^k_{\text{BUF}}; \\
L(P^{j+1}_{\text{BUF}}) &\leftarrow L(P^j_{\text{BUF}}) - L(P^k_{\text{BUF}}); \\
P^i_{\text{HCI}} &\leftarrow P^k_{\text{BUF}}; \\
j &\leftarrow j + 1; \\
i &\leftarrow i + 1; \\
k &\leftarrow k + 1;
\end{align*}
$$

(3.3)
3.4 Performance analysis

In formula (3.3), the first three lines are the segmentation operation of the P-RASP and the reassembly packet shall be converted to normal HCI packets. The fourth line allows normal operations to be adopted. Therefore, the P-RASP is transparent during packet exchange relative to the HCI and the L2CAP layer.

3.4 Performance analysis

First and foremost, the purpose of the P-RASP is to reduce the number of slot operations of the Bluetooth controller and to enhance the efficiency of packet transmission. When the packets are queued in the Bluetooth controller buffer during the idle/sleep interval duration, the LM has enough operation time for reassembly. If the packets are reassembled, then the link activation time can be reduced to improve efficiency. Consider the case where twelve 3-DH1 packet payloads with an additional 2 byte header can be reassembled to one 3-DH5 packet payload, the link activation time is reduced from twenty-four slots (12 transmit slots and 12 ACK slots) to six slots (5 slots for the 3-DH5 and 1 slot for the ACK).

Secondly, the overhead of P-RASP is very low and only adds a two byte payload header for each re-assembled packet. The formulas (3.2) and (3.3) are easy to implement. This cross-layer optimization cost is very low compared for example with the cross-layer approach to send commands repeatedly in the hold mode.

Furthermore, this protocol transparently exchanges the L2CAP PDUs and it doesn’t affect other operations, e.g. encryption, in the LMP, or other functionality, e.g. flow control or error control and retransmissions, in L2CAP. Therefore, security and reliability aren’t compromised.

Last but not least, the P-RASP can be applied widely in Bluetooth-based WSNs. The P-RASP is currently limited to run at the idle/sleep interval duration in the Bluetooth node as happens for example in the sniff mode of participating nodes. Moreover, if there are more than two Bluetooth nodes in the piconet, the idle/sleep interval opportunities frequently appear while the Bluetooth piconet channel is occupied by a pair of nodes and other nodes are waiting to communicate. The appropriate length of idle/sleep interval can be set to improve the performance of the P-RASP and thereby optimize the channel utilization.
3.5 Simulation evaluation

The simulation module of this chapter uses UCBT (University of Cincinnati - Bluetooth), which is a NS-2 based Bluetooth network module and which has been partially updated to specification 2.0 [53]. In the following subsections, first-off, four appropriate scenarios are proposed that have been set up to evaluate the main performance parameter of P-RASP, which is the active time in a Bluetooth link. Secondly, the P-RASP performance versus a set of different packet types and idle/sleep intervals for the scenarios is evaluated. Next, data slot efficiency is explored, and a new metric is introduced known as the data slot efficiency improvement ratio (DSEIR) in order to compare the packets efficiency in Bluetooth. In the final subsection, the P-RASP is evaluated for a piconet with enabled QoS.

3.5.1 Appropriate scenarios

The appropriate application scenarios for P-RASP are often constant bit rate (CBR) services with low data rates. Therefore, representative scenarios with low CBR have been identified as follows:

- **Scenario 1**: Monitoring Electrocardiograph (ECG). A portable ECG sensor generally sends the data payload, which is not encapsulated, with 2 bytes per sample. The sampling rate is in the order of 128 samples / sec [16, 62].

- **Scenario 2**: Monitoring physical activity events. The sensor often uses accelerometer based sensors to determine various activities and events, e.g. fall. The sampling rate for an accelerometer can be in the order of 200 samples per second where each sample is 6 bytes, resulting in a raw data rate of 9.6 kbps.

- **Scenario 3**: Smart watch with GPS function. The data rate for GPS is typically 1 message per second and it uses 4800 baud [63].

When the resources of the device or peer device are restricted, the device’s MTU has to be set to a smaller size. The Bluetooth L2CAP has to segment the SDU to smaller sized PDUs and transmit each separately to peer. As a result, the smaller size baseband packets are used. For example, a smart watch shall transmit mobility and GPS data to a central server which records the location and mobility...
3.5. **Simulation evaluation**

of a monitored person, for example, a patient walking in a garden. The L2CAP MTU of the device is set to 48 bytes due to the watch having inadequate memory to cache the packets at the host. The adopted scenario is as follows.

- **Scenario 4**: A master and seven slaves are formed into a piconet with QoS guaranteed services used between devices. This scenario addresses the P-RASP running on a piconet.

Scenarios 1 and 2 above are assumed to be TCP/IP applications and the Bluetooth Network Encapsulation Protocol (BNEP) [64] is used to encapsulate the application data packets, instead of the related Bluetooth profiles. In addition, as the P-RASP is assumed to run at the idle/sleep interval in the Bluetooth node, the sniff mode will be used.

### 3.5.2 Analysis of active time

For Scenario 1, we set the sniff interval, the $T_{\text{sniff}}$ to 0.4 seconds and the sniff attempt and timeout parameters are set to their minimum values of “1” and “0” respectively (assume latency is irrelevant, and an ideal channel for each link). The simulation results will be used to compare the accumulated active slots per anchor point in the Bluetooth link for the following four cases: 1) the Bluetooth BR controller without P-RASP; 2) the Bluetooth EDR controller without P-RASP; 3) the Bluetooth BR controller with P-RASP; 4) the Bluetooth EDR controller with P-RASP.

At the sniff mode’s anchor point in case 1, when the active link is stable while it has an ideal channel, the Bluetooth BR controller will deliver 53 DH3 packets with a baseband packet payload of 29 bytes (2 bytes for ECG data, 27 bytes in packet header for UDP and L2CAP). Packets are delivered during the sniff attempt duration at case 1. The simulation shows that the number of active slots are 214 slots per anchor point. Similarly, the number of active slots in cases 2, 3 and 4 are 108, 36 and 14 slots per anchor point, respectively. At the sniff mode’s anchor point in case 4, the total of 14 active slots are used, which correspond to two 3-DH5 packets with two associated POLL slots and a POLL-NULL pair for running out the sniff timeout parameter in the sniff mode. The two 3-DH5 have payloads of 992 bytes and 620 bytes, respectively (992 bytes result from the 32 reassembled small baseband packet payloads with 2 bytes for each header, and 620 bytes result from 20 reassembled small baseband packet payloads with 2 bytes for each header).
3.5. Simulation evaluation

Each of these two 3-DH5 packets has a new 2 bytes header with new LLID code ‘00’. Fig. 3.7 shows the accumulated active slots per anchor point in Bluetooth link during transmission time for each case in Scenario 1.

The left side and right side of the histogram in the figure show the required active slots per anchor point while the Bluetooth BR/EDR controller without P-RASP, or with P-RASP in Scenario 1, respectively. Compared with the histograms, the graph indicates that the active slots are reduced by the P-RASP.

In Fig. 3.7, the Bluetooth EDR controller without using the P-RASP protocol requires 7.7 times more active slots than the Bluetooth EDR controller using the P-RASP protocol, and the Bluetooth BR controller without using P-RASP protocol requires 5.9 times more active slots than the Bluetooth BR controller using the P-RASP protocol. Fewer active slots mean lower power consumption in the device. Hence, the P-RASP reduces the active time and improves power consumption of the Bluetooth link.

3.5.3 Set packet types and idle/sleep interval to improve P-RASP efficiency

Scenario 2, which uses TCP/IP to transfer physical activity measurements, has higher data rate than Scenario 1 and sets the $T_{\text{sniff}}$ to 5 seconds, as the application can accept much larger packet delay. The device in Scenario 2 will cache more data than Scenario 1 during the sniff interval, which means it needs a larger buffer. The simulation result is nearly the same as Scenario 1, which is shown in Fig. 3.8.
3.5. **Simulation evaluation**

The four lines in the figure show the accumulated active slots in Bluetooth link during transmission time for each case in Scenario 2 at the anchor point. The anchor points are 5 s, 15 s and 20 s.

Comparing the histograms, the graph indicates that the active slots are reduced by the P-RASP.

**Fig. 3.8:** Accumulated active slots in Bluetooth link during transmission time for each case in Scenario 2.

In **Fig. 3.8**, we clearly observe using accumulated active slots over four anchor points that the use of P-RASP can reduce the active time at each anchor point, where the traffic is basically the same under ideal channel conditions. Furthermore, different preferred packet types were investigated in Scenario 2 with the P-RASP active at the anchor point, and the simulation results are shown in **Fig. 3.9**.

Comparing the histograms, the use of a 3-DH5 packet in Scenario 2 with the P-RASP has minimum active time and maximum transmission efficiency.

**Fig. 3.9:** The number of active slots at anchor point in Scenario 2 with different preferred packet types.
3.5. Simulation evaluation

In Fig. 3.9, the use of a 3-DH5 packet in Scenario 2 with the P-RASP has minimum active time and maximum transmission efficiency. The use of 2-DH1 packet is the least transmission efficient because the 2-DH1 payload is not fully utilized in Scenario 2.

The transmission efficiency can be improved further by optimizing the sniff parameters. By setting the $T_{\text{sni}} = 0.4$ sec in Scenario 1, the Bluetooth EDR controller with P-RASP transmits two 3-DH5 packets at the sniff mode’s anchor point. With maximum delay allowed in this case, the $T_{\text{sni}}$ interval can be adjusted to transmit either one or two 3-DH5 packets with full payloads at each of the sniff mode’s anchor points.

Similarly, Scenario 3, which transfers GPS data, also needs to set the precise $T_{\text{sni}}$ and preferred packet type due to the restricted resources of the device, in which the L2CAP SAR shall segment and reassemble the GPS message, as the MTU is 48 bytes. The GPS message is 4800 baud and the device needs to transmit 480 bytes data per second (when using UART communication to transmit 1 byte, 10 bit are needed—1 start bit, 8 data bits and 1 stop bit). The L2CAP will segment it to ten L2CAP PDUs and the resultant payload of the HCI ACL packets is 50 bytes. We set the different $T_{\text{sni}}$ and the simulation result is shown in Fig. 3.10.

Comparing the histograms, when set $T_{\text{sni}} = 0.5$ s in Scenario 1, the packet transmission in the Bluetooth baseband just uses two 3-DH5 packets with full payload and the efficiency of packets transmission is maximized.

(a) Scenario 1 with different $T_{\text{sni}}$ at anchor point
3.5. Simulation evaluation

Comparing the histograms, when $T_{\text{sniff}} = 1 \text{ s}$, Scenario 3 needs one 3-DH3 packet or two DH5 packets to complete the data transmission; when $T_{\text{sniff}} = 2 \text{ s}$, Scenario 3 needs two 3-DH3 packets or one 3-DH5 packet and one 3-DH1 packet to complete data transmission. Therefore, using $T_{\text{sniff}} = 1 \text{ s}$ and 3-DH3 packet in Scenario 3 maximizes the efficiency of packet transmission.

Fig. 3.10: The comparison of different sniff intervals and preferred packet types for Scenario 1 and 3 with P-RASP

Various simulations were performed by varying the sniff interval and preferred packet, and extracted results are shown in Fig. 3.10.

In Fig. 3.10(a), when we set $T_{\text{sniff}} = 0.5 \text{ s}$ in Scenario 1, the packet transmission in the Bluetooth baseband just uses two 3-DH5 packets with full payload and the efficiency of packets transmission is maximized.

In Fig. 3.10(b), when $T_{\text{sniff}} = 1 \text{ s}$, Scenario 3 needs one 3-DH3 packet or two DH5 packets to complete the data transmission; when $T_{\text{sniff}} = 2 \text{ s}$, Scenario 3 needs two 3-DH3 packets or one 3-DH5 packet and one 3-DH1 packet to complete data transmission. Therefore, using $T_{\text{sniff}} = 1 \text{ s}$ and 3-DH3 packet in Scenario 3 maximizes the efficiency of packet transmission.

3.5.4 Comparison of DM and DH packets by DSEIR

The DH packets on the EDR controller only use a CRC which is effective on low error rate channels. The DM packets on the BR controller often are the only option on high error rate channels, for example, due to interference. Set $\alpha$ to be the packet retransmission ratio per slot ($0 \leq \alpha \leq 0.2$). We can define a parameter to compare efficiency which is called the data slot efficiency improvement ratio (DSEIR) as
3.5. Simulation evaluation

follows.

\[
\text{DSEIR} = \frac{P_{\text{payload}}}{P_{\text{occupied slots}}} (1 - P_{\text{occupied slots}} \times \alpha)
\]  

(3.4)

The DSEIR’s unit is bytes / slot. This parameter can be used to analyze the advantage of selecting packet types in Bluetooth. The calculation results of DSEIR for the baseband packet types are shown in Fig. 3.11.

![Fig. 3.11: Data slot efficiency improvement ratio (DSEIR) in Bluetooth.](image)

The graph indicates that the 3-DH5 and 3-DH3 packets have higher DSEIR for a low error rate channel; The DM3 and DM5 packet have relatively high DSEIR for a high error rate channel.

In Fig. 3.11, the 3-DH5 and 3-DH3 packets have higher DSEIR for a low error rate channel on the EDR controller. However, when the channel has a high error rate, the DM1, DM3 and DM5 packets have to be used. The DM3 and DM5 packet have relatively high DSEIR.

3.5.5 P-RASP in piconet with QoS option

Scenario 4 is a piconet with one master and seven slaves. The polling scheme of piconet is E-limited [25, 65] and the latency in the QoS descriptor is 700 ms. Therefore, the master can set the transmission window of each slave to 100 ms (about 160 slots and therefore realize the polling interval \( T_{\text{poll}} = 700 \text{ ms} \) for each slave). As stated before, the master shall poll each slave at least every 700 ms, and after that, if there is not traffic between the master and the slave, the master sends NULL packets to the slave who need not respond.
3.5. Simulation evaluation

If there are low data rates between the master and the slaves, the P-RASP shall play an important role in the piconet. The buffered small size baseband packets shall be re-assembled by the P-RASP and the Bluetooth device shall use higher DSEIR packets during the data transmission windows. Comparison of the maximum throughput and the actual active time for a slave in the piconet between the P-RASP and normal transmission are shown in Fig. 3.12.

The continuous “-” and “-“ lines in the figure Fig. 3.12 (a) and (b) show the maximum throughput and the actual active time for a slave in the piconet between the P-RASP and normal transmission. From the graph Fig. 3.12(a), the P-RASP increases the maximum throughput of each slave during the limited transmission slots; in Fig. 3.12(b), the active time of a slave in the piconet is reduced in most packet cases by the P-RASP.

In Fig. 3.12a, the P-RASP increases the maximum throughput of each slave during the limited transmission slots. The maximum throughput of lower DSEIR packet types can be overcome using the P-RASP.

In Fig. 3.12b, the active time of a slave in the piconet is reduced in most packet cases by the P-RASP. When the buffered packets are 3-DH3 with full payloads, the 3-DH3 packets will be re-assembled to a 3-DH5 packet by the P-RASP due to the 2 bytes additional packet header. The active time has actually increased in this case. If the buffered packets types are 3-DH3, 2-DH5 and 3-DH5, the DSEIR of these packets are high and it’s unnecessary to re-assemble to a higher one. The payload of 3-DH3, 2-DH5 and 3-DH5 packet are such that the packets cannot be re-assembled with each other. Thus, the P-RASP shall only re-assemble smaller sized buffered packets where the payload of the incoming buffered packet is no larger than 2-DH3’s. This is implemented as a new restriction to packet
reassembly procedures in subsection 3.3.3 and the re-assembly formula (3.2). The additional P-RASP operation can be as follows:

- If the types of current buffered HCI ACL packet $P_{HCI}^{i}$ are 3-DH3, 2-DH5 and 3-DH5, then the arriving packet will be processed as normal.

## 3.6 Summary and further discussion

There are more and more devices embedded with Bluetooth for low data rate applications. When the rate is quite low and much less than the capacity of Bluetooth, the BR/EDR controllers have to do many polling operations to maintain channel synchronization and deliver the data using small size baseband packet type. Many low power operations, which generally reduce polling and save power, result in the Bluetooth devices having an idle/sleep interval duration.

The P-RASP runs during any idle/sleep interval duration in the Bluetooth BR/EDR controller and is designed to reassemble the buffered small size HCI ACL packets to larger ones, so that the Bluetooth LM can assemble a larger baseband packet type (e.g. 3-DH5) with full payload. Using the additional identifier—LLID code ‘00’ in the baseband packets payload header, the P-RASP shall allow the peer Bluetooth device to identify and segment the reassembled packet to restore the original packets.

The P-RASP feature can be negotiated by a new LM command or by using the ATT and GATT, and implemented in the Bluetooth BR/EDR controller. Relative to the HCI and the L2CAP layer, the P-RASP is transparent during packet exchange. By the simulation of several scenarios, the performance analysis reveals that this protocol can significantly reduce the active slots and enhance the efficiency of packet transmission with lower overhead. Based on the data rate and the maximum delay tolerated by a particular application and by setting the parameters of the sniff mode, an ideal idle/sleep interval duration can be set in the controller and allow the P-RASP to fully use the baseband packet payload. In addition, it is recommended that the P-RASP use higher DSEIR packets types as the preferred re-assembled packet types, which are the DM3 and DM5 packets on the Bluetooth BR controller and the 3-DH3 and 3-DH5 packets on the Bluetooth EDR controller. In addition, the P-RASP shall only re-assemble small size buffered packets (up to 2-DH3) given that the DSEIR values are already high for 3-DH3 or larger packet types.
3.6. Summary and further discussion

For many real applications, with energy restrictions, there is a strong requirement to optimize the Bluetooth transfer overhead. Many existing applications already use low power modes or new approaches for autonomously reducing polling and saving power. Therefore, the usage of P-RASP can be adopted in conjunction with these techniques to develop even more efficient packet transmission in the future.
Chapter 4

An Adaptive State-Transition Approach for Power Saving

In Chapter 2, the analysis showed the default polling system in Bluetooth cannot handle several sequential POLL-NULL pairs; and the three low power modes often cannot meet the requirement of applications in view of the difficulty of setting parameters. A new approach is required that can adaptively reduce the unnecessary polling operations and be easy to set parameters.

In this chapter, an adaptive state-transition approach is recommended to reduce the polling operations by setting three different polling intervals, which can be small, medium and large in the controllers. Each controller runs a common algorithm to choose among the three polling intervals and adaptively transfers link state from active to idle while maintaining system synchronization. The state machine and state-transition rules of this state-transition approach are given and its performance and parameter setting will be analyzed in detail.

In order to study this proposed state-transition approach, it is necessary to build a simulation tool to illustrate the performance achieved. In the simulation section within this chapter, the simulation framework is presented in detail. The simulation results will be examined in comparison to the default polling system and the standard defined Sniff mode.

The chapter is structured as follows. Section 4.1 presents the proposed new approach, and its state machine diagram and state-transition rules. The performance analysis is described in Section 4.2. Section 4.3 is the simulation based evaluation. Finally, Section 4.4 presents the summary and further discussion of this research.
4.1 Adaptive state-transition approach

The current low power operations can reduce the polling operations by entering an idle/sleep state. However, when the applications are changed, the parameters of current low power operations might be improper and need to be changed based on the applications. The parameters of the low power operations are not easy to set to cater for the requirements of the variety of applications. Therefore, the new approach is required to adaptively reduce polling operations with more flexible for setting the parameters. The new approach will be first outlined by giving an overview of the technical operation.

4.1.1 The kernel idea of reduced polling operation

Polling operations, i.e. the exchange of POLL-NULL packets, can be reduced to allow the controllers on both devices to enter the idle or sleep state. The peer controllers in an active connection run a common algorithm to transfer the state between the active state and the idle or sleep state which automatically adapts and minimizes the polling interval while ensuring that data transfer is efficient. The kernel idea is shown in Fig. 4.1.

![Fig. 4.1: Peer controllers run a common algorithm to reduce polling operations.](image_url)

In Fig. 4.1, the peer devices on a connection run a common algorithm that reduces the number of POLL-NULL packet pairs, through a shared set of states and by transferring the communicating devices’ state between active and idle or sleep using state transition rules.
4.1. Adaptive state-transition approach

4.1.2 State machine diagram

Before proposing a set of states and state transition rules, a Finite State Machine (FSM) diagram is used to represent the common algorithm introduced in Fig. 4.1. The proposed state-transition approach’s state machine diagram is shown in Fig. 4.2.

The polling interval, which is the time between two active states in Bluetooth, can have a very wide range of choices, but will be shown to be simplified if the common polling interval can be “small”, “medium” and “large”. Therefore, the basic concept is to introduce four main states: an active state, and a state for each polling interval.

Furthermore, when the devices are in the Bluetooth CONNECTION state, packets can be exchanged. The new approach is entirely supported within the CONNECTION state. The state machine within the CONNECTION state consists of four states. The active sub-states in the active state are interim states that are used to monitor the traffic and decide if the current state remains as the active state or changes to one of the small, medium or large interval states. This state machine diagram is informative and doesn’t represent the following scenario. In the case of nothing received during the active sub-states, the next state is always...
4.1. Adaptive state-transition approach

the active state.

4.1.3 Proposed approach’s state-transition rules

The proposed approach’s state-transition rules in each Bluetooth controller on a specific connection for ACL applications is as follows:

- If the Bluetooth controller is in the **active state** and transmits or receives a data packet, the Next State shall be the **active state**;

- If the Bluetooth controller is in the **active state** and detects a polling operation, the Next State shall transfer to the **small interval state** for a common “small” interval duration;

- If the Bluetooth controller exits the **small interval state** and completes a polling operation at the **active state**, the Next State shall transfer to the **medium interval state** for a common “medium” interval duration;

- If the Bluetooth controller exits the **medium interval state** and completes two sequential polling operations at the **active state**, the Next State shall transfer to the **large interval state** for a common “large” interval duration;

- If the Bluetooth controller exits the **large interval state** and completes two sequential polling operations at the **active state**, the Next State shall transfer to the **large interval state** for a common “large” interval duration;

- In addition, if the Bluetooth controller is in the **active state** and has detected nothing or incorrect packet at the **active state**, the Next State shall be the **active state** and the packet re-transmission will be processed by the LM till successfully send or the link loss is detected.

The rules above use 1 or 2 sequential polling operations to determine the next state of the controller, which will be one of the small, medium or large interval states based upon the received POLL-NULL and the current state. Using 2 sequential polling operations prevents link synchronization loss by wireless channel interference or piconet collisions during a large sleep interval. The common algorithm shall run on both controllers. The values of the “small”, “medium” and “large” intervals on the controllers can be a set of default values or they can be set and tuned separately at the start of each active link by cross-layer negotiation for example with a new LM command. Later default values will be proposed.
4.2 Performance analysis

Fig. 4.3 shows the proposed state-transition approach’s operations, when there is not data traffic to transmit over an active link.

4.2 Performance analysis

This section investigates the performance of the proposed approach in comparison to both the conventional polling system and the sniff mode, which have been presented in Chapter 2.

4.2.1 Parameter setting

As discussed in Chapter 2, the default polling system can’t easily adapt the polling interval and the parameters of sniff mode are quite difficult to set and use for the varied applications. Relative to these two approaches, the parameters of three polling intervals of the proposed approach are easy to set.

The small interval can be used to reduce several POLL-NULL pairs and to ensure a minimum delay on an active link, e.g. when the controllers are waiting for the response of LM command or the availability of the buffer in the controllers. This analysis will start by recommending values for the three polling intervals. The value of small interval is set to 8 slots, which is on the larger side of the generally selected smaller polling intervals, which are typically between 2 and 10 slots. This is a trade-off between packet delay and the frequency of state-transitioning to the medium interval state.

The medium interval will be set equal to 38 slots: this corresponds to the idle slots in the default polling system, which is customarily used on almost all the...
4.2. Performance analysis

Bluetooth links.

The large interval can be set to the maximum acceptable delay of the application or can be set equal to the sniff interval parameter of Sniff mode. Using the interval of Sniff mode, if available reduces, repetitive design of the large intervals in the same application.

4.2.2 Context analysis

First and foremost, the proposed approach has three polling intervals to be selected that cater for the varied application scenarios from light traffic to heavy traffic. The proposed approach can transfer the state on the controller to the small idle interval state to reduce the need for continuously polling operations, which often happens on many of the existing controllers when bench observed. The state-transition to the medium interval only needs to detect the POLL-NULL operation twice—one is to transfer to the small interval and the other is to transfer to medium interval, which is a significant improvement over the default polling system. When the proposed approach transfers to the large interval state on the controllers, the actual polling period slots will have included a small interval plus a medium interval. Compared with the difficulty of setting sniff attempt and interval parameters in Sniff mode, the parameter setting of the proposed approach is much easier and more flexible.

Secondly, the link synchronization loss is also considered. This approach doesn’t affect the existing packet error handling in Bluetooth. When the master sends a POLL and doesn’t receive a packet from the slave, the link synchronization loss is detected by the master. Recovery from a loss of synchronization is possible such that re-transmissions are carried out until the link supervision timer results in a timeout. The proposed approach uses different observations for different polling intervals. The two sequential POLL-NULL pairs, which are used before the start of the large sleep interval state in the controllers, is to ensure the master has received the NULL packets to defend against occasional wireless channel interference or piconet collisions.

Last but not least, the kernel algorithm of the approach is a traffic-aware algorithm and is consistent with reinforcement learning [66]. When the controllers receive a POLL-NULL pair, they can have a common idle interval as reward. The more rewards the controllers have, the longer the idle interval they can have. The learning system architecture is shown in Fig. 4.4. This can lead to much more
complicated and adaptive reward schemes.

Fig. 4.4: The proposed state-transition approach’s reinforcement learning system architecture.

4.2.3 Comparison with existing polling system and sniff mode

As discussed, the common idle intervals of the proposed approach can be set “small” = 8 slots, “medium” = 38 slots and “large” = 796 slots (approximately half a second corresponding to a typical sniff sleep interval). Respectively, the polling system and the proposed approach’s operations in the active link without traffic are illustrated in the previously presented figures: Fig. 2.1 and Fig. 4.3.

In the polling system, when Bluetooth establishes an ACL link with no active traffic from the upper layer, the link with the default polling system will typically have 40 poll operations per second after 10 sequential POLL-NULL operations as required by the polling system specification. In this link without data traffic, using the proposed approach in the controllers the polling operations will be 4 poll operations per second when stable. When the Bluetooth link has variable active traffic, the controllers can reduce small intervals of unnecessary polling operations by adopting a state-transition to the small idle interval. Therefore, the proposed approach shall be more efficient than the existing polling system.

In Sniff mode, the Bluetooth devices agree periodic anchor points where they will communicate. Consequently, some parameters need to be negotiated, e.g. a sniff interval, sniff attempt and sniff timeout. The master shall only start a transmission to the slave in the specified sniff attempt slots, and the slave may return to sleep in the remaining slots of the sniff interval period [35]. Fig. 4.5 gives
4.2. Performance analysis

an overview of the operations of the slave transmitter and receiver, further details are presented in Section 2.4, Chapter 2.

For comparison, set sniff interval = 800 slots, sniff attempt = 4 slots and sniff timeout = 2 slots. If there is not active traffic, the controllers in Sni mode shall resynchronize with two POLL-NULL operations at the sniff attempt slots and go to sleep for 796 slots; If there is data transmission at master-to-slave slots, the slave shall continue listening for sniff timeout slots, therefore any buffered data can be transmitted from the master to slave as the sniff timeout counter will restart in the slave when a data packet is received; however, if there is data transmission at slave-to-master slots, the slave shall continue listening sniff timeout = 2 slots but the master may decide not to transmit a packet and consequently the slave has to wait for the next anchor point to transmit data.

The proposed approach is more flexible and can achieve the low power result of Sni mode, by adopting the three “sni sleep interval” equivalents which are “small” = 8, “medium” = 38 and “large” = 796 slots. In general, the “small”, “medium” and “large” parameters of the proposed approach in Bluetooth can be set for the equivalent sni requirements. Furthermore, the approach hasn’t the sniff timeout problem as the state-transition rules claim that both controllers shall continue to transmit or receive packets if they have received a data packet.

If the sniff parameters—sni attempt and sni timeout adopt other values in Sni mode, the proposed approach also can achieve the same or even a better result, through the design of the hidden states and by redefining the observations in the system model with the corresponding parameters.
4.3 Simulation evaluation

This section shall present an assessment of the proposed approach and compare it with the polling system and Sniff mode.

The simulation of Bluetooth scenarios often uses Blueware [52] or UCBT (University of Cincinnati - Bluetooth) [53], which are NS-2 based Bluetooth network modules. These two modules implement a near complete Bluetooth (specification 1.1 or 2.0) stack, and include support of the Bluetooth Baseband, LMP, and host layers such as L2CAP. The traffic of a Bluetooth scenario is created by Tcl (Tool Command Language) scripts. However, the created traffic by Tcl usually just contains one or two representative application scenarios.

To analyse the proposed approach, it is only necessary to represent the packet pairs between the controllers, and there is no benefit to introduce the full packet processing of a Bluetooth protocol stack. The traffic of the validation Bluetooth scenarios can be represented by recording the types of packet pairs in the wireless channel. In the following subsections, firstly several simulation scenarios will be identified, then the simulation setup will be described and simulation results will be presented.

4.3.1 Simulation scenario structures and traffic observation sequence

A typical traffic pattern in Bluetooth is bursty traffic, e.g. a file transfer, periodic or aperiodic low rate traffic, e.g. from a sensor. However, even though Bluetooth applications are varied, from the perspective of the observer, the Bluetooth wireless channels normally only have two types of packet pairs:

- The data packet pairs which includes four different patterns: Data-NULL pairs, POLL-Data pairs, NULL-Data pairs, and Data-Data pairs.
- The POLL-NULL pairs.

To define a scenario, the symbol “1” is used to indicate one of the data packet pairs and the symbol “0” a POLL-NULL pair. A set of “1” and “0” symbols defines an observation sequence of traffic which can be used to represent the activities on a Bluetooth wireless channel. A simulation scenario is illustrated in Fig. 4.6.

The interval between data packet pairs and POLL-NULL pairs depends on many factors, for example, whether the traffic is constant bit rate (CBR) or
variable bit rate (VBR); whether it requires packet segmentation and reassembly; the packet flow control and process time in the buffer, etc. For simulation purposes, the rand(1, len) function in MATLAB can be used to produce a set of “0” and “1” symbols with a specified probability, e.g. the “0” probability is \( p \) and “1” probability is \( r \) in the sequence, to emulate the activity on a Bluetooth channel. The specified “1” probability \( r \) determines the number of data packet slots and reflects the specified data traffic throughput. The code for an observation sequence of the simulation scenarios is shown in Table 4.1.

Table 4.1: The code for an observation sequence of simulation scenario.

<table>
<thead>
<tr>
<th>Step</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>len=1000000;</td>
<td>Set the observation sequence length;</td>
</tr>
<tr>
<td>2</td>
<td>( p = 1 - r );</td>
<td>Set the “0” probability ( p ) in the sequence;</td>
</tr>
<tr>
<td>3</td>
<td>seq=(rand(1,len));</td>
<td>Produce random numbers between 0 and 1;</td>
</tr>
<tr>
<td>4</td>
<td>seq(find(seq &gt; p))=1;</td>
<td>“1” with specified probability ( r );</td>
</tr>
<tr>
<td>5</td>
<td>seq(find(seq &lt;= p))=0;</td>
<td>“0” with specified probability ( p );</td>
</tr>
<tr>
<td>6</td>
<td>dlmwrite('seq.txt', seq);</td>
<td>Record simulation scenario.</td>
</tr>
</tbody>
</table>

Appendix A shows more detailed simulation code of an observation sequence generation.

The created traffic from Table 4.1 shall have various intervals with consecutive sequences of “0”, which the controller could use to enter an idle or sleep state. The traffic between the Bluetooth controllers can be light, or heavy. The specified “1” probability \( r \) in Table 4.1 can be set as 0.1 or smaller for light traffic and 0.9 or bigger for heavy traffic. Here, the research is simplified by defining a data packet as only occupying 1 slot (in fact they can also be of 3 or 5 slots duration in the Bluetooth standard).
4.3. Simulation evaluation

4.3.2 Simulation setup

In [67], in order to collect simulation data to carry out a trace analysis, a physical simulator based on MATLAB has been developed. The Bluetooth simulation module of this paper uses MATLAB to create an observation sequence of traffic, for numerical analysis and for evaluation of the implementation of proposed approach. Therefore, the proposed approach is run for the defined scenarios presented in the last subsection 4.3.1 by the state-transition rules and Viterbi algorithm and is evaluated using comparisons with the polling system and Sniff mode. We assume the channel interference is irrelevant and adopt an ideal channel for each Bluetooth link. All the simulation takes place when the three approaches are running over an active link. Although the created Bluetooth scenarios don’t simulate an actual Bluetooth wireless channel between the controllers, they cover the majority of circumstances in an actual channel with specified traffic throughput and with various idle intervals.

4.3.3 Simulation results

Based on the simulations, the average power, the average end-to-end delay, and the number of occupied slots of each state of the three approaches are analyzed and compared next. In order to optimize the state parameter settings, the proposed approach is further evaluated for alternative states with different small interval parameters. Appendix B shows more detailed simulation code of the proposed state-transition approach.

A. Average power

The power used in the Bluetooth controllers includes the transmit packet power, the receive packet power and idle power. Based on a Bluetooth chip data sheet [59], the normalized power values are considered as follows:

\[ P_{avg}(\text{Data packet pairs}) = 1 \text{ unit per slot}; \]
\[ P_{avg}(\text{POLL - NULL pairs}) = 0.8 \text{ unit per slot}; \]
\[ P_{avg}(\text{Idle}) = 0.01 \text{ unit per slot}; \]

And the average power of the polling system and sniff mode can be calculated by the followed equation.

\[ P_{avg}(Q) = \frac{\sum P_{avg}(\text{State})*\text{(the state's slots)}}{(\text{Total slots})} \] (4.1)
4.3. Simulation evaluation

In different throughput traffic conditions with one million observation sequences, and using equation (4.4), the average power’s simulation results for the three approaches—the polling system, Sniff mode and the proposed approach in the Bluetooth controllers are shown in Fig. 4.7.

![Graph showing average power comparison](image.png)

The continuous “-,” “-” and “-” lines in the figure show the average power when the polling system, Sniff mode and the proposed state-transition approaches is running at the defined scenarios with offered traffic from light to heavy in the Bluetooth wireless Channel, respectively. The graph indicates that the average power of state-transition approach will be less than the polling system’s and slight more than Sniff mode’s.

From Fig. 4.7, under the conditions specified and comparing the proposed approach and polling system, the average power is significantly reduced, and hence there is the opportunity for significant power saving, when the proposed approach is used, especially for the low data rate applications—where the probability of data packets is close to 0.1 or smaller. At light traffic of 0.1, the average power is reduced by a factor of 3.6, compared to the current polling system.

Comparing the proposed approach and Sniff mode, the proposed approach will consume more power than Sniff mode. The reason is the proposed approach has to transition through the small and medium intervals before getting the benefit of the large interval idle/sleep time, in this case using the non-ideal interval periods “small” = 8, and “medium” = 38. If the intervals “small” and “medium” are all set to 796 and we re-set the observation \( v_2 \) as equivalent to \( v_3 \), and Sniff mode parameters—sniff attempt and timeout are set to 2 (which means the observation of Sniff mode are the same with the proposed approach), the simulation result show the proposed approach’s average power is the same as Sniff mode’s for the same
4.3. Simulation evaluation

scenario. Therefore, the proposed approach can be considered as an alternative to Sniff mode by setting the intervals appropriately.

B. Average end-to-end packet delay

The three approaches all introduce additional packet end-to-end delay when they transfer to the idle interval while a data packet is ready for transmission. The end-to-end packet delay $d_{\text{end-end}}$ includes transmission delay, propagation delay and processing delay.

\[ d_{\text{end-end}} = d_{\text{trans}} + d_{\text{prop}} + d_{\text{proc}} \quad (4.2) \]

\[ d_{\text{avg,trans}} = \frac{\sum (d_{\text{trans}} \text{for each packet})}{(\text{Total packets})} \quad (4.3) \]

The propagation delay and processing delay are ignored, since it’s typically very small in Bluetooth. The maximum packets transmission delay of the three approaches equate to the polling or sniff interval. For example, Sniff mode maximum packet delay is about 796 slots, which is 497.5 ms as each slot is 625 $\mu$s in Bluetooth. Under the same observation sequences of traffic as in the last subsection, Fig. 4.8 shows the simulation results for the average $d_{\text{end-end}}$ comparison by using equations (4.2) and (4.3).

![Graph showing comparison of average end-to-end packet delay](image)

(a) The approach vs. Polling system

The continuous “-” and “.-.” lines in the figure show the average end to end packet delay when the polling system and the proposed state-transition approaches is running at the defined scenarios with offered traffic from light to heavy in the Bluetooth wireless Channel, respectively. The graph indicates that the average end to end packet delay of state-transition approach will be slightly higher than the polling system’s.
4.3. Simulation evaluation

(b) The approach vs. Polling system

The continuous “-○-” and “-●-” lines in the figure show the average end to end packet delay when Sniff mode and the proposed state-transition approaches is running at the defined scenarios with offered traffic from light to heavy in the Bluetooth wireless Channel, respectively. The graph indicates that the average end to end packet delay of state-transition approach will be much lower than Sniff mode’s.

Fig. 4.8: Average end-to-end packet delay comparison of three approaches in different traffic conditions in Bluetooth wireless channel.

The simulation in Fig. 4.8 shows that the polling system has the minimum average packet delay and Sniff mode has very large average packet delay, especially in heavy traffic. The main reason is Sniff mode can’t change the sniff parameters after setup and can only select one sniff interval. If the traffic changes, the inappropriate sniff interval shall result in the larger packet delay.

Compared with the existing polling strategy and Sniff mode, as shown in Fig. 4.8(a) and Fig. 4.8(b), the $d_{\text{avg,trans}}$ of the proposed approach is less than 12 ms/packet when the traffic is light and is less than 2 ms/packet when the traffic is heavy. Related to light traffic and taking into account power saving, the 12 ms/packet delay is generally acceptable. In relation to heavy traffic, the proposed approach’s packet delay is far lower than Sniff mode.

C. Number of occupied slots of each state

The Bluetooth controller can be characterized with two kinds of states: active (delivery of packets, $q_1$) and idle for a particular interval (the idle interval of polling system or Sniff mode, $q_2$). The proposed approach’s states are shown in Fig. 4.2.
4.3. Simulation evaluation

The states in Fig. 4.2 can be defined as follows:

\[ Q = \{q_1, q_2, q_3, q_4\} = \{q_1 = \text{active state}; q_2 = \text{idle/sleep with “small” common interval state}; q_3 = \text{idle/sleep with “medium” common interval state}; q_4 = \text{idle/sleep with “large” common interval state}; \} \] (4.4)

Fig. 4.9 shows the occupied slots of each state of the three approaches for four traffic simulation scenarios, which is defined by setting the data packet probability parameter \( r \) in the observation sequence.

![Histograms showing the number of occupied slots for different traffic conditions.](image)

Fig. 4.9: Number of occupied slots of each state of three approaches in different traffic conditions in Bluetooth wireless channel.

The simulation results show that the proposed approach has less active slots (the state \( q_1 \)) than the polling system, especially when low data traffic throughput occurs as shown in Fig. 4.9(a). From Fig. 4.9(a-d), Sniff mode always has more idle slots, which means lower power consumption but with longer packet delay. From Fig. 4.9(a), the proposed approach has more slots at the state \( q_4 \) when data traffic is low. From Fig. 4.9(d), when traffic is heavy, the proposed approach has only a few slots at the state \( q_2 \) and no slots at either the state \( q_3 \) or \( q_4 \). This means the proposed approach adjusts the states according to the traffic levels. Hence, the
4.3. Simulation evaluation

The proposed approach improves on both the existing polling system and Sniff mode, with less active slots and shorter packet delay, resulting in significant advantage.

D. Alternative state parameters scenario

In this analysis, various interval sizes were explored, for example, in the case below by setting the small interval to 4, 8 and 18 slots. The medium and large intervals are left unchanged.

The simulation results for the average power and end-to-end delay are shown in Fig. 4.10(a) and Fig. 4.10(b). As expected, the average power is reduced by the increase in small interval duration as shown in Fig. 4.10(a). Less obvious, from Fig. 4.10(b), is that the shorter small interval increases the frequency of state-transition and the controller shall transfer quicker to medium and large interval which causes a longer packet delay at light traffic. The heavy traffic often has the intermixed POLL-NULL pairs which can be reduced by transitions to the shorter small interval idle period with a shorter packet delay. Therefore, the shorter small idle interval results in the shorter packet delay for heavy traffic.

The defined simulation scenarios are random sequences with defined throughput traffic rates with varied intervals, which don’t represent any particular Bluetooth application on the wireless channel. An actual Bluetooth scenario often has regular
4.4. Summary and further discussion

Fig. 4.10: Different small idle interval of proposed approaches in different traffic conditions in Bluetooth wireless channel.

4.4 Summary and further discussion

The study proposed a new strategy for reducing power consumption by improving the polling operation. The new approach uses a set of three different polling intervals in the Bluetooth BR/EDR controllers, whereby the controllers can adaptively choose the intervals and link state transfers from active to idle based on a common algorithm. This approach can be implemented autonomously on the controllers and significantly improve Bluetooth power efficiency by reducing the polling operations. A set of default parameters can be used, and it can easily be added to Bluetooth as a feature. In fact, similar to the polling system, once the controllers are aware that their peer supports it in the default scenario, it can be automatically activated without any application level involvement. If necessary, additional configuration options can be supported through new HCI and/or LMP commands.
4.4. Summary and further discussion

To illustrate the performance achievement, simulation results are examined in comparison to the default polling system and the standard defined Sniff mode. The simulation results showed that although the power consumption of the proposed approach is slightly more than adopting Sniff mode for a typical configuration, it has very low average end-to-end packet delay and is easier and more flexible for setting the parameters. The state-transition approach adjusts the states according to the traffic levels. The state setting can be optimized for a specific Bluetooth scenario.

The simulation scenarios are based on random sequences with defined throughput traffic rates and variable data packet intervals. Many applications are much more deterministic with further opportunity for overall optimization and tradeoffs between power saving and average or worst-case end-to-end delay. Detailed channel modelling was not required to compare the proposed approach with the polling system and the Sniff mode, as only the packet pairs between the controllers needed to be considered. In particular, the algorithm handles errors in the transmission by staying in the active state, hence not justifying bit or packet level error processing.

However, the three polling intervals with the proposed parameter setting may not attain the best of power saving for an application. The kernel algorithm of the approach is a traffic-aware algorithm. When we have the characteristics of a particular application, the challenge is to propose improved parameter settings or state-transition rules and algorithms cater for it. In the next chapter, further modeling analysis of this approach is undertaken in order to develop more and better common state-transition algorithm in the future.
Chapter 5

The proposed state-transition approach’s system modelling and model analysis

A problem of fundamental interest is characterizing real-world signals, e.g. the state-transition in the Bluetooth controllers, in terms of signal models \[44\]. In the last chapter, an adaptive state-transition approach has been put forward for power saving. The analysis in subsection 4.2.3 at Chapter 4 reveals that the proposed approach is more flexible and has three “interval” equivalents, which can achieve the low power result of Sniff mode.

In order to further study improvements parameter settings or the improved setting of state-transition rules and algorithms, this chapter establishes a system model of the proposed state-transition approach by the Hidden Markov model (HMM). The HMM properties can be used for further analysis, e.g. to estimate state-transition paths, to calculate the average power and to design a broadly applicable common algorithm for the controllers. The new state-transition algorithm for a special application can be designed by a HMM. The polling system and Sniff mode with defined observations can be analysed by the HMM.

The chapter is structured as follows. Section 5.1 presents the proposed state-transition approach’s system modelling. Section 5.2 includes the model analysis and parameter estimation and the model utilization is described in Section 5.3. Finally, Section 5.4 presents the summary and further discussion of this research.
5.1 System modelling

The author in [45] provides a simple analytical model to compute the saturation throughput performance of the 802.11 using Markov chains. The state-transition in the Bluetooth controllers is a stochastic process and also is a Markov chain. This is because the possible states in Bluetooth controllers are finite and independent from each other. The traffic from applications is also a stochastic process. A Hidden Markov Model (HMM) is a double stochastic process with one underlying process that is not observable but may be estimated through a set of processes that produce a sequence of observations [46]. Thus, this section establishes a system model of the proposed approach based on the classic HMM.

As mentioned in Chapter 4, the controllers have two major states on an active ACL link in Bluetooth, one is to be active—the delivery of packets (includes data packets and POLL-NULL packets) and the other is to be idle or sleep with an agreed common interval. In this idle or sleep state, power saving can be achieved by turning off many subsystems, i.e. the receiver and various hardware clocks. Furthermore, the transition from one state to the next is a stochastic process with the Markov property, where the next state depends only on the current state with constant probabilities. A Markov matrix can be used for the state-transition between the active state and the idle/sleep interval states in Fig. 4.2 as follows:

\[
\begin{bmatrix}
 p_{11} & p_{12} \\
 p_{21} & p_{22}
\end{bmatrix}
\]

The entries of the matrix are the state-transition probabilities.

The state-transition in the controllers is not observable by the peers and can’t be directly determined by the peer controller. Therefore, the state-transition in the controllers is “hidden”. The various traffic between the controllers is another stochastic process, which is observable by the controllers. Although the state-transition in the controllers can’t be directly determined, the observable sequence by the controllers, which is the traffic (the exchanged packet types—either data packets or POLL-NULL packets), provides the probabilistic information regarding the state in the controllers. Therefore, this state-transition issue in Bluetooth can be described with the classic HMM [44, 46, 47].

The following notation for describing the HMM will be used [47]:
5.1. System modelling

\[ T = \text{the length of the observation sequence;} \]
\[ N = \text{the number of states in the model;} \]
\[ M = \text{the number of observation symbols;} \]
\[ Q = \{q_1, \ldots, q_N\}, \text{the set of states of the Markov process;} \]
\[ V = \{v_1, \ldots, v_M\}, \text{the set of possible observations;} \]
\[ A = \text{the matrix of state transition probabilities;} \]
\[ B = \text{the observation probability matrix;} \]
\[ \pi = \text{the initial state distribution;} \]
\[ O = (O_1, \ldots, O_T), \text{the observation sequence;} \]
\[ X = (X_1, \ldots, X_T), \text{the hidden states sequence.} \]

In terms of the state-transition rules mentioned in Chapter 4, there are four hidden states \( X \) in the controllers during an active connection as follows: the transmission and reception of packets state, and the common idle/sleep period states with “small”, “medium” or “large” interval.

Therefore, set \( N = 4 \) and the simplified states \( Q \) in the Bluetooth controllers are as follows:

\[
Q = \{q_1, q_2, q_3, q_4\} = \{q_1 = \text{active state}\};
\]
\[ q_2 = \text{idle/sleep with “small” common interval state;} \]
\[ q_3 = \text{idle/sleep with “medium” common interval state;} \]
\[ q_4 = \text{idle/sleep with “large” common interval state;} \]

(5.1)

Set the number of observation symbols \( M = 3 \), which means that the set of possible observations \( V \) is:

\[
V = \{v_1, v_2, v_3\} = \{v_1 = \text{Data packet;} \]
\[ v_2 = \text{POLL – NULL;} \]
\[ v_3 = \text{POLL – NULL twice}; \}

(5.2)

Assume the initial state distribution as \( \pi = (1, 0, 0, 0) \), this means that the controller starts in the active state. The state transition probability matrix \( A = \{a_{ij}\} \) is \( N \times N \) matrix with \( a_{ij} = P(\text{state } q_j \text{ at } t+1|\text{state } q_i \text{ at } t) \), as shown in Table 5.1:

\(^1\)The \( q_1 \) refers to the active sub-state 1 in Fig. 4.2. The active sub-states 2 and 3 are intermediate states between the \( q_1 \) state and the \( q_2, q_3 \) or \( q_4 \) states and they are considered to be part of the state-transition rules.
\(^2\)Need to observe over four slots while the current state is \( q_3 \) or \( q_4 \).
5.1. **System modelling**

Table 5.1: The state transition probability matrix \(A\) and corresponding states.

<table>
<thead>
<tr>
<th>Current state (q_i)</th>
<th>(q_1)</th>
<th>(q_2)</th>
<th>(q_3)</th>
<th>(q_4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(q_1)</td>
<td>(a_{11})</td>
<td>(a_{12})</td>
<td>(a_{13})</td>
<td>(a_{14})</td>
</tr>
<tr>
<td>(q_2)</td>
<td>(a_{21})</td>
<td>(a_{22})</td>
<td>(a_{23})</td>
<td>(a_{24})</td>
</tr>
<tr>
<td>(q_3)</td>
<td>(a_{31})</td>
<td>(a_{32})</td>
<td>(a_{33})</td>
<td>(a_{34})</td>
</tr>
<tr>
<td>(q_4)</td>
<td>(a_{41})</td>
<td>(a_{42})</td>
<td>(a_{43})</td>
<td>(a_{44})</td>
</tr>
</tbody>
</table>

The observation probability matrix \(B = \{b_{jk}\}\) is an \(N \times M\) matrix with \(b_{jk} = P(\text{observation } k \text{ at } t \mid \text{state } q_j \text{ at } t)\), as shown in Table 5.2:

Table 5.2: The observation probability matrix \(B\) and corresponding states.

<table>
<thead>
<tr>
<th>Observation symbol (v_k)</th>
<th>(v_1) = Data packet</th>
<th>(v_2) = POLL – NULL</th>
<th>(v_3) = POLL – NULL twice</th>
</tr>
</thead>
<tbody>
<tr>
<td>Next states (q_j)</td>
<td>(q_1)</td>
<td>(b_{11})</td>
<td>(b_{12})</td>
</tr>
<tr>
<td></td>
<td>(q_2)</td>
<td>(b_{21})</td>
<td>(b_{22})</td>
</tr>
<tr>
<td></td>
<td>(q_3)</td>
<td>(b_{31})</td>
<td>(b_{32})</td>
</tr>
<tr>
<td></td>
<td>(q_4)</td>
<td>(b_{41})</td>
<td>(b_{42})</td>
</tr>
</tbody>
</table>

The \(a_{ij}\) and \(b_{jk}\) are the state-transition probabilities which are illustrated in Fig. 5.1.

Fig. 5.1: The state-transition in the BT controllers with the probabilities \(a_{ij}\) and \(b_{jk}\) based on the HMM.
5.2. The model analysis and parameter estimation

Therefore, the system model of the proposed state-transition approach is a classic HMM $\lambda$, which can be described as follows:

$$\lambda = (X, O, \pi, A, B)$$

(5.3)

5.2 The model analysis and parameter estimation

Based on the state-transition rules and the system model described in subsection 4.1.2 and subsection 4.1.2, this section shall further analyze and estimate the values of the state transition probabilities in the system model. The current polling system will be analyzed at first, and then the proposed state-transition approached will be addressed.

5.2.1 The current polling system modelling

The current polling system in Bluetooth can be modeled by the HMM. In the case of point to point connection with the default poll interval, it has two states in the controllers under the assumption that the master user the full Poll-Null between retransmission, which are:

$$Q = \{q_1 = \text{active state}, q_2 = \text{idle with 38 slots interval state}\}$$

(5.4)

and has three possible observations which is:

$$V = \{v_1 = \text{Data packet},$$
$$v_2 = \text{ten sequential POLL – NULLs},$$
$$v_3 = \text{POLL – NULL}\}$$

(5.5)

The observation probability matrix $B$ is:

$$B = \{b_{jk}\} = \begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 1
\end{bmatrix}$$

(5.6)

As Fig. 5.2 shows, based on the possible observations and direct estimation method, the state-transition in the controller is unchanged and the estimation of observation probabilities are constant values, which have the probabilities “1” and

$^3$Only need to observe two slots if the observation is $v_3$ while the current state is $q_2$. 
5.2. The model analysis and parameter estimation

“0” in equation (5.6).

![Diagram of the state-transition of the current polling system based on a HMM](image)

Fig. 5.2: The state-transition of the current polling system is based on a HMM with the probabilities “1” and “0”.

From Fig. 5.2, when the Bluetooth link is at the current state \( q_1 \) and after ten polling operations, the observation \( v_2 \) happens and the Bluetooth controllers shall transfer to the next state \( q_2 \). If the observation \( v_3 \), which is a single polling operation happens in the state \( q_2 \), the next state shall be also the state \( q_2 \), which allows power saving, in particular in the master. The state transition probability matrix \( A \) in the model is as follows:

\[
A = \{a_{ij}\} = \begin{bmatrix}
a_{11} & a_{12} \\
 a_{21} & a_{22}
\end{bmatrix} \tag{5.7}
\]

In equation (5.7), the parameters of transition probabilities cannot be estimated directly due to the variety of applications. The value of state transition probabilities in \( A \) reflect the characteristics of the applications and it doesn’t affect the state-transition rules of the polling system. The observation probability matrix \( B \) in equation (5.6) has reflected all the state-transition rule of the polling system.

### 5.2.2 The model parameter estimation

First of all, consider the system model example of the polling system presented in previous subsection, and recalling that there are four hidden states and three observations in the proposed approach, the parameters of the corresponding system model can be estimated to be the probabilities “1” or “0”.

Next, the controller’s state-transition rules in subsection 4.1.3 can be explained as follows, where the next state \( X_{i+1} \) is determined by the current state \( X_i \) and
5.2. The model analysis and parameter estimation

the observation $O_i$:

- When the $O_i = v_1$, the $X_{i+1} = q_1$ with probability “1”;
- When the $O_i = v_2$ and $X_i = q_1$, the $X_{i+1} = q_2$ with probability “1”;
- When the $O_i = v_2$ and $X_i = q_2$, the $X_{i+1} = q_3$ with probability “1”;
- When the $O_i = v_3$ and $X_i = q_3$, the $X_{i+1} = q_4$ with probability “1”;
- When the $O_i = v_3$ and $X_i = q_4$, the $X_{i+1} = q_4$ with probability “1”;
- When the $O_i = $ Nothing or packet incorrect, the $X_{i+1} = q_1$ with probability “1”;

The first rule says that when the observation is a data packet then the next state is the active state. The second rule indicates that if the observation is a single POLL-NULL polling operation occurring in the active state, then the next state is the idle/sleep small interval state. The other rules further show the probability relations between the observations and next states. This will give the observation probability matrix $B = \{b_{jk}\}$.

Moreover, the entries of state transition probability in matrix $A = \{a_{ij}\}$ also reflect the state-transition rules in addition to the characteristics of the application. From Fig. 4.2, some state-transitions also can’t occur in the controllers due to the rules, e.g. when the active state is in the active sub-state 1, the probabilities of state-transition from $q_1$ to $q_3$ and from $q_1$ to $q_4$ is “0”. The other two examples are: (1) when the active state is in the active sub-state 2, the probabilities of state-transition from $q_2$ to $q_2$ and from $q_2$ to $q_4$ is “0”; (2) when the active state is in the active sub-state 3, the probabilities of state-transition from $q_3$ and $q_4$ to $q_2$ and from $q_3$ and $q_4$ to $q_3$ is “0”. This will give the state transition probability matrix $A = \{a_{ij}\}$.

Therefore, the matrices $A = \{a_{ij}\}$ and $B = \{b_{jk}\}$ in the HMM for the new approach are

$$A = \begin{bmatrix} a_{11} & a_{12} & 0 & 0 \\ a_{21} & 0 & a_{23} & 0 \\ a_{31} & 0 & 0 & a_{34} \\ a_{41} & 0 & 0 & a_{44} \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.8)$$
5.3. The model utilization

The non-zero values in the state transition probability matrix \( A = \{a_{ij}\} \), which reflects the characteristics of the application, doesn’t affect the state-transition rules in the controllers.

Finally, from the analysis above, the proposed approach in Chapter 4 can be modeled as a HMM with the states in equation (5.1), the observations in equation (5.2), the initial state distribution \( \pi \) and the parameters \( A = a_{ij} \) and \( B = b_{jk} \) in equation (5.8).

5.3 The model utilization

The established system model is a HMM, therefore the HMM properties can be utilized. The HMM often is used to solve three types of basic problems [44, 47], which are as follows.

Problem 1: Given the model and the observation sequence, how to compute the probability of the observation sequence.

Problem 2: Given the model and the observation sequence, how to choose a corresponding hidden state sequence which is optimal.

Problem 3: Given an observation sequence, how to adjust the model parameters to maximize the probability of the observation sequence.

Problem 1 and 3 are used to evaluate, design and optimize the model and its parameters. Solving Problem 2 is our concern in this chapter as it’s used to estimate state-transition paths, resulting in the state-transition algorithm.

5.3.1 State-transition path estimation

Based on equation (5.8) and the state statistics for a specified application, a generic HMM equation (5.9) \( \lambda = (\pi, A_1, B_1) \) with probabilities could be obtained and one such scenario could be:

\[
A_1 = \begin{bmatrix}
0.2 & 0.8 & 0 & 0 \\
0.5 & 0 & 0.5 & 0 \\
0.9 & 0 & 0 & 0.1 \\
0.9 & 0 & 0 & 0.1 \\
\end{bmatrix}, \quad \text{and} \quad B_1 = \begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1 \\
\end{bmatrix}
\]

(5.9)

and \( \pi = \{1,0,0,0\} \). Given the observation sequence \( O = (O_1, O_2, ..., O_T) \), the challenge is to find the “optimal” hidden state sequence \( X = (X_1, X_2, ..., X_T) \).
5.3. The model utilization

The Viterbi Algorithm [44, 48] exploits recursion to reduce computational load, and uses the context of the entire observation sequence \( O \) to recapture the “optimal” state sequence \( X \) with computational complexity \( N^2T \) [48]. For example, when the observation sequence \( O \) is as follows:

\[
O = (v_1, v_2, v_3, v_3, v_1, v_2, v_2, v_1, v_1, v_2, v_2, ...) \]

And using the Viterbi Algorithm, the state-transition of the controllers is most likely as follows:

\[
X = (q_1, q_2, q_3, q_4, q_4, q_1, q_2, q_3, q_1, q_1, q_2, q_3, ...) \]

This can be implemented using the hmmviterbi(\( O, A_1, B_1 \)) function in MATLAB [54]. The output of hidden states \( X \) is governed by the state-transition rules in subsection 4.1.3 at Chapter 4.

The direct computation of a generic HMM in the Bluetooth controllers by the Viterbi Algorithm is generally infeasible due to the restrictive computational load. However, the adoption of values “1” and “0” for the state-transition probabilities in the system model greatly simplifies the calculation. In addition, given an observation sequence \( O \) in the system model, the state-transition path in the controllers can also be determined by the state-transition rules. Therefore, the non-zero values in the state transition probability matrix \( A = \{a_{ij}\} \), which could be obtained by the state statistics for a specified application, doesn’t affect the state-transition rules in the controllers if the observation sequence \( O \) are unchanged.

The implementation of the system model in the controller is relatively easy. Nevertheless, the system model in equation (5.8) can be used to calculate the average power and to design the state-transition rules in the Bluetooth controllers.

5.3.2 Average power calculation

Given the observation sequence \( O \) for a particular Bluetooth active link and the state-transition rules, this active link’s state transition probability matrix \( A \) can be acquired by statistical analysis. The active link’s average power for the proposed approach can be solved by linear equations (5.10) and equation (5.11) as follows.

\[
(A' - I_4) \times \begin{bmatrix} p(q_1) \\ p(q_2) \\ p(q_3) \\ p(q_4) \end{bmatrix} = 0 \tag{5.10}
\]
5.3. The model utilization

\[ P_{\text{avg}}(Q) = \sum_{i=1}^{4} p(q_i) \times P_{\text{avg}}(\text{State}(q_i)) \]  (5.11)

The \( I_4 \) in linear equations (5.10) is the identity matrix and the equation (5.10) can be extended as follows:

\[
\begin{align*}
(a_{11} - 1) \times p(q_1) + a_{21} \times p(q_2) + a_{31} \times p(q_3) + a_{41} \times p(q_4) &= 0 \\
a_{12} \times p(q_1) - 1 \times p(q_2) + 0 \times p(q_3) + 0 \times p(q_4) &= 0 \\
0 \times p(q_1) + a_{23} \times p(q_2) - 1 \times p(q_3) + 0 \times p(q_4) &= 0 \\
0 \times p(q_1) + 0 \times p(q_2) + a_{34} \times p(q_3) + (a_{44} - 1) \times p(q_4) &= 0
\end{align*}
\]

In equations (5.10) and equation (5.11), the \( p(q_i) \) is the probabilities of being in each state in a particular active link; the \( P_{\text{avg}}(\text{State}(q_i)) \) is the average power of each state. For example, set \( P_{\text{avg}}(\text{State}(q_1)) = 1 \) and set all the others to 0; and take the specific example given by the HMM of equation (5.9), the \( p(q_i) \) and average power \( P_{\text{avg}}(Q) \) in the active link are:

\[
(p(q_1), p(q_2), p(q_3), p(q_4)) = \left( \frac{45}{101}, \frac{36}{101}, \frac{18}{101}, \frac{2}{101} \right)
\]

\[ P_{\text{avg}}(Q) = \sum_{i=1}^{4} p(q_i) \times P_{\text{avg}}(\text{State}(q_i)) = \frac{45}{101} \]

If the active link stays in the active state, the average power \( P_{\text{avg}}(Q) \) by the assumption above:

\[ P_{\text{avg}}(Q) = P_{\text{avg}}(\text{State}(q_1)) = 1 \]

The active link can state-transition from \( q_1 \) to \( q_2 \), \( q_3 \) and \( q_4 \) by the proposed approach, the average power \( P_{\text{avg}}(Q) \) in the specific active link is down to \( \frac{45}{101} \).

5.3.3 New state-transition rules design

For a specific Bluetooth scenario with the states \( Q \) in the controllers and possible observations \( V \) defined in Section 5.1, the state-transition rules and common algorithm in the controllers can be designed by setting the entries in the matrices \( A \) and \( B \).

For instance, consider the scenario of periodic low data rate applications which have small intervals mainly and haven’t large intervals, the new state-transition rules could be set to transfer to the state \( q_1 \), \( q_2 \) and \( q_3 \), and not transfer to the state \( q_4 \). By this new state-transition rules, the parameters \( A = \{a_{ij}\} \) and \( B = \{b_{jk}\} \) in
5.3. The model utilization

the HMM are

\[
A = \begin{bmatrix}
a_{11} & a_{12} & 0 & 0 \\
a_{21} & 0 & a_{23} & 0 \\
a_{31} & 0 & a_{23} & 0 \\
0 & 0 & 0 & 0 \\
\end{bmatrix}, \quad \text{and} \quad B = \begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 1 & 0 \\
0 & 0 & 0 \\
\end{bmatrix}
\]

(5.12)

On the contrary, if the applications can accept very large packet delay, the new state-transition rules could be set to only transfer to the state \(q_1\) and \(q_4\) thereby minimizing power consumption. Therefore, the parameters \(A = \{a_{ij}\}\) and \(B = \{b_{jk}\}\) in the HMM for this new state-transition rules are

\[
A = \begin{bmatrix}
a_{11} & 0 & 0 & a_{14} \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
a_{41} & 0 & 0 & a_{44} \\
\end{bmatrix}, \quad \text{and} \quad B = \begin{bmatrix}
1 & 1 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 1 \\
\end{bmatrix}
\]

(5.13)

Moreover, each state \(q_i\) and possible observation \(v_i\) in the system model also can be re-designed, if needed. Using the equations (5.4) - (5.7), the polling system is one of many scenarios that the system model supports. The state-transition of the low power sniff mode also can be represented by this system model with corresponding states and observations. We can set the states \(Q\) and the possible observations \(V\) for Sniff mode as follows.

\[
Q = \{q_1 = \text{active state}, q_2 = \text{idle till next anchor point state}\}
\]

(5.14)

It has three possible observations which

\[
V = \{v_1 = \text{Sniff attempt is NOT 0}, \quad v_2 = \text{Sniff attempt is 0 and Sniff timeout is NOT 0}, \quad v_3 = \text{Sniff attempt is 0 and Sniff timeout is 0}\}
\]

(5.15)

The observation probability matrix \(B\) is estimated as follows.

\[
B = \{b_{jk}\} = \begin{bmatrix}
1 & 1 & 0 \\
0 & 0 & 1 \\
\end{bmatrix}
\]

(5.16)
5.4. Summary and further discussion

The value of the state transition probability matrix $A$ for the sniff is as follows.

$$ A = \{a_{ij}\} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \quad (5.17) $$

The $a_{ij}$ reflect the characteristics of the applications and doesn’t affect the state-transition rules of Sniff mode. Therefore, using the equations (5.14) - (5.17), Sniff mode can be presented by the HMM mode. The corresponding system model utilization analysis can therefore be applied.

5.4 Summary and further discussion

The proposed state-transition approach in Chapter 4 is a polling scheduling scheme and can adaptively improve the polling operation in Bluetooth. This chapter gives the mathematical description of the state-transition approach.

To begin with, we set the states on the controllers and a set of possible observations between the controllers while using the proposed state-transition approach. And then, the system model is established based on the classic Hidden Markov Model. The state transition probability matrix $A$ and the observation probability matrix $B$ are given.

Then, the model’s parameters $A$ and $B$ were obtained by the state-transition rules. The HMM properties can be used for further analysis of the proposed state-transition approach. We used the established system model to estimate state-transition paths on the controllers and to calculate the average power of the proposed approach for a generic example. Furthermore, we can use the established system model to design new state-transition rules for special applications. The observations can also be re-designed by the designers with the HMM.

Finally, this chapter provides the current polling system’s model by the HMM, which is in expression (5.4), (5.5) and (5.6), and the sniff mode’s model by the HMM, which is in expression (5.13), (5.14), (5.15) and (5.16). Thus, as a conclusion, the HMM can be a common model to analyze the state-transition issues and be used to design and develop more efficient low power mode in Bluetooth.

In addition, overall, this chapter focus on the mathematical modelling and the model utilization. The HMM may aid research and development teams to model, analyze and design a new low power operation in Bluetooth at a system level. The limitations of this research that it lacked the simulation of an
5.4. *Summary and further discussion*

actual Bluetooth application scenarios and just analyzed point to point network. Therefore, within the proposed state-transition approach, future work can focus on some target applications, e.g. Bluetooth-based sensor networks, by performing further assessment of the analytical model in these actual Bluetooth scenarios. A more deeper comparison with adaptive polling schemes such as E-limited will be undertaken and piconet based scenarios will be addressed.
Chapter 6

M/G/1/N Queueing Model and Hold mode for Cross-layer Approach

The analysis in Chapter 2 has discussed that Hold mode is designed to “run only once” for each invocation and the previous mode is restored after the Hold time. The Bluetooth specification doesn’t give specific guidance for the repeated use of Hold mode. In this chapter, the research proposes using Hold mode repeatedly to transmit data in Bluetooth, in particular for applications such as those frequently accessing in a Bluetooth-based medical body area network (BT-MBAN).

The characteristics and requirements analysis of BT-MBAN indicate that the rate of data transfer in a medical body area network (MBAN) is low, for most application scenarios, compared to the capabilities of Bluetooth. As such, the Bluetooth polling system wastes energy and hence a cross-layer interaction is proposed based on queueing theory and Hold mode in Bluetooth. The employment of an M/G(M/M)/1/N queueing model is a guidance for the repeated use of Hold mode in Bluetooth, which can improve power efficiency significantly for data transmission in many MBAN scenarios.

The chapter is structured as follows. Section 6.1 includes related analysis about this proposal. The core design is described in Section 6.2. Section 6.3 is the M/G(M/M)/1/N Queueing Model in Bluetooth and Section 6.4 gives detailed operation analysis. The performance analysis of this research design is discussed in Section 6.5, which includes the model analysis and the parameters optimization of the BT-MBAN system by means of the Queueing model. Finally, Section 6.6
6.1. Related analysis

In addition to understanding the new research design for power saving in this chapter, we will analyze a few key questions: (1) the application’s Characteristics and Requirements; (2) Hold mode review and (3) the comparison with Sniff mode while using Hold mode repeatedly.

6.1.1 Characteristics and Requirements Analysis of BT-MBAN

Many medical applications of wireless body network are described in [68, 69]. From these medical applications, the special characteristics and requirements of a BT-MBAN scenario can be inferred as follows.

First of all, although the applications of medical sensor are varied, the actual bit rates of MBANs are far below the capacity of a Bluetooth network. For example, the sustainable raw data rates of a fall and activity sensor [22], which is a medical monitoring device, is 9.6 kbps and many other devices have significantly lower aperiodic data rates, e.g. ECG [16, 62]. The Bluetooth BR/EDR supports asynchronous connections with data rates of 721.2 kbps for Basic Rate, and 2.1 Mbps for Enhanced Data Rate, which is enough but the problem is ineﬃciency.

Next, long data delays in MBAN can be tolerated (seconds to minutes range) and generally accepted, but low delay constraints are speciﬁed when an emergency happens.

The third is based on network conﬁguration; the MBAN topology is point to point or point to multi-point and it does not dynamically change. Therefore, the topology is simple and Bluetooth’s piconet meets the requirements in general (but is typically constrained to a master and up to seven active slaves).

Last but not least, the central node of a MBAN might have an uninterruptible power supply or high capacity battery, but the slave nodes usually just have a low capacity battery for power supply. Therefore, to reduce the power consumption and prolong the usable time of slave nodes, each slave node should just acquire and transmit the data and the central node shall process the data in most cases.
6.1. Related analysis

6.1.2 Bluetooth Hold mode review

In Chapter 2, we have a brief summary of Hold mode in Bluetooth. Hold mode often is designed as a network scheduling method for the piconet or scatternet in Bluetooth. When the devices enter the Hold time of Hold mode, the current ACL logical transport in the active link is paused and the power consumption of devices is reduced. This section shall detail the HCI & LMP commands and the operations of Bluetooth devices.

The host can use a HCI Hold Mode command to place a device into Hold mode. The Controller may do this by either negotiating Hold mode parameters or forcing Hold mode. Hold mode will automatically end after the negotiated length of time. Note the HCI Hold Mode parameters include a min and max Hold interval, allowing the LMP a degree of flexibility in selecting the Hold time (Hold duration). An example of a Hold mode request is shown in Fig. 6.1. The optional LMP_set_AFH command can be used to adapt the set of active frequencies hopping channels.

![Fig. 6.1: Hold request](image)

To enter Hold mode, both the master and slave can start a request or can force Hold mode through the LMP commands. The process is initiated by sending an LMP_hold_req PDU or LMP_hold PDU containing two parameters—Hold time and Hold instant. The LMP sequence for a master to force a slave into Hold mode is shown at Fig. 6.2.

From specification, the LMP_hold and LMP_hold_req PDUs both contain a parameter, Hold instant, that specifies the instant at which the hold becomes
6.1. Related analysis

![Diagram of Master and Slave in Hold mode]

Fig. 6.2: Master forces slave into Hold mode.

effective. The Hold instant is chosen by the sender of the message and should be at least $6 \times T_{\text{poll}}$ slots in the future. $T_{\text{poll}}$, which is set by the LM, is defined as the maximum time between transmissions from the master to a particular slave on the ACL logical transport.

6.1.3 Using Hold mode repeatedly and the comparison with Sniff mode

Hold mode can be re-used by the host by sending HCI & LMP commands again, as it’s “run only once” for each invocation. For a low rate BT-MBAN, Sniff mode is often used by the designers. The difference between Hold mode (repeated use) and Sniff mode are as follows.

Firstly, during the Hold time of Hold mode, the Bluetooth device has more power save than during the idle time of Sniff mode. This is because the ACL logical link is paused in Hold mode and the device needn’t process the traffic incoming from applications but buffer it. However, the ACL logical link is available during the idle time in Sniff mode. Therefore, the devices have to process the traffic incoming from applications as normal during Sniff mode, even if the transmitter (Tx) and the receiver (Rx) are not active.

Secondly, the cost of entering each mode is different. The entering Hold mode needs $6 \times T_{\text{sniff}}$ slots as Hold instant for each invocation. The default value of $T_{\text{sniff}}$ is 40 slots. It can be 6 slots to affect hold as minimum, or 240 slots as default. While waiting for the Hold instant slots, ACL packets can also be delivered. Therefore, the cost of entering Hold mode can be very few slots for LMP commands if the Hold instant slots are also used for delivery data. The entering Sniff mode is determined by the current CLK and sniff parameters ($D_{\text{sniff}}$) satisfy the applicable equations which are in the specification. Generally, the cost of entering Sniff mode is nearly a $T_{\text{sniff}}$ period, which can be used for data delivery but often is just used for polling operations. This may far exceed the cost of entering Hold mode.

Thirdly, the sniff parameters are unchanged during Sniff mode. The hold
6.2 New design to use Hold mode in BT-MBAN

parameters can be changed each time a hold request is made. This makes Hold mode more flexible but for the expense of the hold parameters being renegotiated.

Finally, as shown in Fig. 6.2, the device can force Hold mode on the peer device, which means we can control the device to enter Hold mode for power saving according to traffic conditions. During Sniff mode, the devices have to run the rules of Sniff mode, which is determined by the sniff attempt and timeout slots. The sniff parameters can’t change according to traffic conditions except by exiting Sniff mode. Thus, if the application data is periodic, the designers have to do a lot of duplicate testing to find the better sniff parameters for different applications; if application data is aperiodic, the designers have to set average sniff parameters, but these may not be optimal resulting in very little control and a waste of energy.

6.2 New design to use Hold mode in BT-MBAN

As mentioned above, the arrival rate of data in a MBAN is slow in most scenarios. Hold mode in Bluetooth can pause the ACL logical link by suspending data traffic and reduce the polling operation between Bluetooth devices which means power saving. Therefore, in order to improve Bluetooth transmission efficiency, a BT-MBAN device can buffer data when Bluetooth enters Hold mode and transmit the buffered data when Bluetooth exits Hold mode, as the long data delays in MBAN can be tolerated. Moreover, the repeated use of a controlled Hold mode result in significant power saving in BT-MBAN. Thus, the new design in a BT-MBAN is as follows.

1. Bluetooth devices buffer data and enter Hold mode with an estimated Hold time $t_h$, which can be the time for the buffer to reach a set threshold.

2. When exiting hold, Bluetooth devices transmit data over an estimated active time $t_a$ and enter hold mode again with an estimated Hold time $t_h$. The $t_a$ shall include the time of data transmission, another hold request and accept request, Hold instant and unexpected other events, e.g. data retransmission.

3. The Hold time $t_h$ and the active time $t_a$ are determined by the application data profiles.

The key of this design is the set of hold parameters and the active time $t_a$ based on the BT-MBAN applications. In order to improve this design and optimize the parameters, queuing theory can be used to analyse the system model.
6.3. M/G(M/M)/1/N Queueing Model in Bluetooth

6.3 M/G(M/M)/1/N Queueing Model in Bluetooth

Queueing theory \[49\] has its roots early in the twentieth century in early studies on telephone networks. Today it is one of the primary tools used to deal with questions involving trade-offs between the amount of resources allocated to provide a telecommunications service and the quality of service \[50\].

Queueing theory is the mathematical study of waiting lines, or queues. In \[51\], the performance of a single Bluetooth piconet is analyzed using the theory of M/G/1/N queues with vacations. The M means the “Markovian” queue is distinguished by a Poisson arrival process and exponential service times. The G means the arrivals are again Poisson but the service times are described by an arbitrary (or general) probability distribution. The ‘1’ indicates the number of servers in the queue. N is the amount of queue space, which means the size of the buffer.

Therefore, the BT-MBAN systems can be also considered as a M/G(M/M)/1/N queueing model. As Fig. 6.3 shows, the data arrival average rate from BT-MBAN applications is denoted \(\lambda\) bps. The service average rate is \(\mu\) bps. The transmit rate of data of Bluetooth has two processes (M/M). Process 1 is \(\mu_h = 0\) bps when Bluetooth enters Hold mode till the buffer’s threshold is reached; Process 2 is \(\mu_a\) bps when Bluetooth exits Hold mode to become active and transmits data till the buffer is empty.

![M/G(M/M)/1/N Queueing Model](image)

Fig. 6.3: M/G(M/M)/1/N Queueing model in Bluetooth.

The variable \(t_h\) is \(\text{holdTO}\) (Hold time) value in process 1 and the variable \(t_a\) is \(\text{activeTO}\) (transmit time) value in process 2, when the BT-MBAN system enters steady state.

When using a average service rate \(\mu\) to replace the two processes, the BT-MBAN can be considered as a M/M/1/N queueing and based on queueing theory \[49, 50\], the system utilization \(\rho\), which is one of the queueing model factors, can
6.3. M/G(M/M)/1/N Queueing Model in Bluetooth

be defined:

\[ \rho = \frac{\lambda}{\mu} = \frac{\lambda(t_h + t_a)}{\mu_h t_h + \mu_a t_a} = \frac{\lambda}{\mu_a} \left(1 + \frac{t_h}{t_a}\right) \quad (\rho \leq 1) \]  

(6.1)

Therefore, the probability that the system is empty, which means the probability that no data transmission takes place during an interval, is \( P(0) \).

\[ P(0) = 1 - \rho = 1 - \frac{\lambda}{\mu_a} \left(1 + \frac{t_h}{t_a}\right) \]  

(6.2)

The expected amount of data \( L \) in system and expected queue length of data \( L_q \) are

\[ L = \frac{\rho}{1 - \rho} = \lambda W, \quad L_q = \frac{\rho^2}{1 - \rho} = \lambda W_q \]  

(6.3)

The total expected waiting time \( W \) in system and the expected waiting time \( W_q \) in the queue are

\[ W = \frac{1}{\mu - \lambda}, \quad W_q = \rho W \]  

(6.4)

We can analyse in detail an example of the use Hold mode in BT-MBAN and calculate the factors of the queueing model. The data arrival average rate from a medical monitoring device, e.g. a fall and activity sensor [22], can be \( \lambda = 9.6 \) kbps. The data rate in process 2 in Fig. 6.3 is \( \mu_a = 2178.1 \) kbps, which is the maximum rate of Bluetooth while using 3-DH5 packets. Set \( t_h = 30 \) s and \( t_a \) is about 240 - 320 slots, which is 0.15 - 0.2 s. Therefore, Set \( t_a = 240 \) slots = 0.15 s, from equations (6.1) - (6.4), the factors of the queueing model are:

\[ \rho = \frac{\lambda}{\mu_a} \left(1 + \frac{t_h}{t_a}\right) = \frac{9.6}{2178.1} \left(1 + \frac{30}{0.15}\right) = 88.59\% \]

\[ P(0) = 1 - \rho = 11.41\% \]

\[ L = \frac{\rho}{1 - \rho} = 7.76; \quad L_q = \frac{\rho^2}{1 - \rho} = 6.88 \]

\[ W = \frac{1}{\mu - \lambda} = 0.81; \quad W_q = 0.71 \]

Set \( t_a = 320 \) slots = 0.20 s, from equations (6.1) - (6.4), the factors of the queueing model are:

\[ \rho = \frac{\lambda}{\mu_a} \left(1 + \frac{t_h}{t_a}\right) = \frac{9.6}{2178.1} \left(1 + \frac{30}{0.20}\right) = 66.55\% \]

\[ P(0) = 1 - \rho = 33.45\% \]

\[ L = \frac{\rho}{1 - \rho} = 1.99; \quad L_q = \frac{\rho^2}{1 - \rho} = 1.32 \]

\[ W = \frac{1}{\mu - \lambda} = 0.21; \quad W_q = 0.14 \]

From the calculation above, the factors \( L, L_q, W \) and \( W_q \) haven’t direct meaning
6.4. The operation analysis

within the BT-MBAN system. The reason the equations (6.2) - (6.4) are based on M/M/1/N model and we assumed that the BT-MBAN system was a M/M/1/N model and used a service average rate $\mu$ to replace the actual two processes. Actually, the BT-MBAN system is a M/G(M/M)/1/N model, the factors $L$, $L_q$, $W$ and $W_q$ in average can’t reflect the BT-MBAN system performance directly.

Here, the equation (6.1), which is the system utilization factor $\rho$, reflects the BT-MBAN system performance. The examples above have shown that the less active time $t_a$ leads to higher system utilization. The subsequent sections will further discuss this queueing model in BT-MBAN.

6.4 The operation analysis

Our operation analysis, as already shown in the previous section, includes Hold mode, application data rate, buffer data and Bluetooth transmit data with different rates, which are good indicators for the operations in a BT-MBAN. The analysis of specific operations between Bluetooth devices are given as follows.

6.4.1 Master’s operation

The master must take more actions in the BT-MBAN. When the master receives the necessary information from the slave in the connection state, it will decide when the slave’s state shall be transferred to Hold mode with Hold time $t_h$ and appropriate Hold instant. The ideal method for master is to set the hold parameters each time by LMP commands. However, the simplest method for the master is to accept the suggested policy or parameters from the slave by the attribute protocol and generic attribute profile, which include the applications info. However, the master must consider the current network topology and activity. For example, if the master has more than one slave in the piconet, it must maintain fairness.

When the slave leaves Hold mode and enters active mode, a new timer shall be initialized with $activeTO$ value (active time $t_a$), which is a new parameter. During the active time, the master shall send another Hold request with the calculated $t_h$ and Hold instant; when the $activeTO$ timer expires, the master shall transfer the slave’s state to Hold mode. The active time $t_a$ shall be decided by the amount of data in the buffer for transmission, the time necessary to send Hold request, response and Hold instant, and current channel conditions with
appropriate redundancy. As described above, Bluetooth devices shall do periodic state-transition between active and Hold mode, which is shown in Fig. 6.4.

Fig. 6.4: Hold mode based approach to transmit data when stable in a BT-MBAN.

From Fig. 6.4, the \( t_h \) and \( t_a \) could be the same values when stable traffic occurs, e.g. Constant Bit Rate, in a BT-MBAN.

### 6.4.2 Slave’s operation

Generally, the slave’s functions in a BT-MBAN are to receive and perform control commands from the master and to complete data collection, storage and transmission to the master. As the slave has energy restrictions, many special operations, for example to initiate Hold mode or process data, shall be dealt with by the master.

Furthermore, the current medical application attributes can be stored in the slave’s generic attribute profile, which contains the normal rate \( \lambda \) and buffer size \( N \), the acceptable maximal data delay \( t_{\text{delay}} \), suggested Hold time parameters \( t_h \) and Hold instant, and active time parameters \( t_a \) etc. When the ACL link between Bluetooth devices is established, the attributes shall be exchanged by attribute protocol. Of course, the parameters will be decided autonomously by the master and the slave’s operations in active mode shall be minimized.

### 6.5 Performance analysis

#### 6.5.1 The parameters analysis by Queueing model

From Fig. 6.4, it can be seen that the key parameters of the system are the Hold time \( t_h \) and the active time \( t_a \). In an actual system, the Hold time \( t_h \) is determined by the BT-MBAN system maximal acceptable data delay \( t_{\text{delay}} \), the buffer size \( N \),
6.5. Performance analysis

and the rate $\lambda$. The $t_h$ is

$$t_h \in [0, \min(t_{\text{delay}}, \frac{N}{\lambda})] \quad (6.5)$$

From (6.1), $t_a$ is

$$t_a = \frac{\lambda t_h}{\rho \mu_a - \lambda} = \frac{t_h}{\frac{\mu_a}{\lambda} - 1} \quad (6.6)$$

The $\mu_a$ is the actual transmit data rate of Bluetooth, which is determined by the transmission packet types selected and current channel conditions. For the DM1, the DM5 and the 3-DH5 packets in Bluetooth, their maximum rate is 108.8 kbps, 477.8 kbps and 2178.1 kbps, respectively.

Fig. 6.5 shows, when the BT-MBAN system utilization $\rho$ is increased, the active time $t_a$ is reduced while the Hold time $t_h$ is unchanged by (6.6).

![Fig. 6.5: The new design’s system active time $t_a$, based on different BT-MBAN system utilization $\rho$ and Bluetooth transmit data rate $\mu_a$, when set $\lambda = 9.6$ kbps, $N = 4$ kbit, $t_h = \frac{N}{\lambda}$.

The continuous “- ▼ -”, “-◆-” and “-▲-” lines in the figure show the new design’s system active time $t_a$ when Bluetooth transmit data rate is 108.8 kbps, 477.8 kbps and 2178.1 kbps respectively. The graph indicates that the larger value of system utilization $\rho$ will lead to smaller system active time and more power saving.

The BT-MBAN system utilization $\rho$ is necessarily less than 1 due to wireless channel re-synchronization and channel errors, hold request and response messages, and polling operations. Actually, all above take up data transmit channel time, which reduces the system utilization and increases the active time $t_a$.

Assume that the average power of Bluetooth in MBAN during $t_h$ and $t_a$ is 0 and $P_{\text{active}}$, respectively. The new design’s power consumption is $P_{\text{active}} \times t_a$. 98
6.5. **Performance analysis**

Considering the sustainable data rates from applications and regardless of the Bluetooth POLL interval, the conventional power consumption during \( t_h \) and \( t_a \) is \( P_{active} \times (t_a + t_h) \).

Due to equation (6.6) the new design’s power consumption saving utilization \( \rho_{saving} \) is defined as follows:

\[
\rho_{saving} = 1 - \frac{P_{active} \times t_a}{P_{active}(t_h + t_a)} = 1 - \frac{\lambda}{\rho \mu_a} \quad (\rho_{saving} \leq 1) \tag{6.7}
\]

From equation (6.7), the \( \rho_{saving} \) is a function of \( \rho, \mu_a \). Fig. 6.6 shows the impact of \( \rho \) and \( \mu_a \) for \( \rho_{saving} \).

![Fig. 6.6: The power consumption saving utilization \( \rho_{saving} \), based on different BT-MBAN system utilization \( \rho \) and Bluetooth transmit data rate \( \mu_a \), when set \( \lambda = 9.6 \) kbps.](image)

The continuous “-.-”, “-○-” and “-●-” lines in the figure show power consumption saving utilization \( \rho \) when Bluetooth transmit data rate is 108.8 kbps, 477.8 kbps and 2178.1 kbps respectively. The graph indicates that the larger value of \( \rho \) will produce larger power consumption saving utilization and more power saving.

If the power consumption saving utilization \( \rho_{saving} \) equates to zero, there is no power saving in BT-MBAN. Therefore, the power control should make the power consumption saving utilization \( \rho_{saving} \) close to 1 as much as possible. In Fig. 6.6, when \( \mu_a = 9.6 \) kbps, the simulation results show the proposed queueing model based design can improve power consumption saving utilization \( \rho_{saving} \), which means saving energy. With the increasing of BT-MBAN system utilization, the power consumption saving utilization is increased and more energy is saved. For the same BT-MBAN system utilization, the higher the data transmission rate, the
more energy can be saved.

6.5.2 Optimization of Parameters by Queueing model

It is very important to reduce the cost of BT-MBAN system by optimizing their parameters. The queueing model is often used to explore and improve the system. The parameters for optimization are the buffer size $N$ and the Hold time $t_h$ in a given BT-MBAN system. In this section, the cost optimization refers to the whole power consumption of BT-MBAN system.

The whole power consumption is one of the important measurements of a BT-MBAN system. The notations are introduced as follows:

- $F$: whole power consumption in BT-MBAN per unit of time; the unit is $\text{Joule/s}$ or $\text{Watt}$.
- $f_1$: power consumption per unit of time when BT-MBAN is actively running; the unit is $\text{Watt}$.
- $f_2$: power consumption per unit of time per $kbit$ when BT-MBAN system is buffering data; the unit is $\text{Watt}/kbit$.
- $f_3$: power consummated when BT-MBAN transfers state from Hold to active; the unit is $\text{Joule}$.

Therefore,

$$F = f_1 + f_2 \times t_h + \frac{f_3}{t_h + t_a}$$  \hspace{1cm} (6.8)

The $\bar{L}$ is the expected amount of data in the buffer. From Fig. 6.4, the BT-MBAN system is a $\text{M/G(M/M)}/1/N$ queueing model and the buffered data is $\bar{L} = \lambda \times t_h$ when the system is stable. Therefore, from equation (6.1), the equation (6.8) is as follows:

$$F = f_1 + \lambda \times t_h \times f_2 + \frac{f_3}{t_h + \frac{\lambda}{\rho \mu_a - \lambda}} = f_1 + \lambda \times t_h \times f_2 + \frac{(\rho \mu_a - \lambda)f_3}{(\rho \mu_a)t_h}$$  \hspace{1cm} (6.9)

Using (6.5) and (6.9), it is possible to calculate the minimum of $F$ by calculating when $\frac{\partial F}{\partial n} = 0$ and $\frac{\partial F}{\partial N} = 0$, or by using numerical methods. Therefore,

$$\frac{\partial F}{\partial t_h} = 0 + \lambda \times f_2 - \frac{(\rho \mu_a - \lambda)f_3}{(\rho \mu_a)} \times \frac{1}{(t_h)^2} = 0$$

That is

$$t_h = \sqrt{\frac{(\rho \mu_a - \lambda)}{(\rho \mu_a) \times \lambda} \times \frac{f_3}{f_2}}$$  \hspace{1cm} (6.10)

Therefore, when the $t_h$ is set to the value in equation (6.10) at a specified
6.5. Performance analysis

BT-MBAN, the $F$ reaches the minimum.

For example, as mentioned before, the sustainable raw data rates of a fall and activity sensor [22], which is a medical monitoring sensor, is $\lambda = 9.6$ kbps. We use DM5 packet ($\mu_a = 477.8$ kbps) to transmit data and set the system utilization as $\rho = 90\%$ due to system redundancy. The task of the fall sensor is monitoring and we set that this MBAN system acceptable maximal data delay $t_{delay} = 30$ s. In addition, we assume that $f_1$ is 0 (or a constant), $f_2 = 1 \times 10^{-4}$, $f_3 = 0.06$. We can numerically analyze equations (6.10) when $t_h \in [0.5, 30]$, which is shown as Fig. 6.7.

From Fig. 6.7, it shows that the whole power consumption per unit of time $F$ can reach the minimum when $t_h$ is equal to about 7.8 s.

Therefore, $\rho = 0.9$, ($\mu_a = 477.8$ kbps) and $t_h = 7.8$ s, from equation (6.1), we can calculate $t_a \approx 0.174$ s $\approx 280$ slots. This BT-MBAN system will have the lowest cost with the minimum power consumption based on above parameter setting.

In addition, using equation (6.9), the buffer size of this BT-MBAN system can be designed for slightly larger than $N = \rho \times t_h = 9.6 \times 7.8 = 74.88$ kbit. This value can be a set threshold. When the threshold is reached, the ACL link shall be affected and the host can transmit a larger service of data units (SDU).
6.6. Summary and further discussion

to the Bluetooth L2CAP. The Bluetooth L2CAP can improve segmentation and reassembly (SAR) with a larger SDU and the baseband can assign larger baseband packet types (e.g. DM5 or 3-DH5) as appropriate. Therefore, the transmit data rate $\mu_a$ in a BT-MBAN could reach the maximum data throughput rate of Bluetooth.

6.6 Summary and further discussion

The chapter proposed the repeated use of a controlled Hold mode to transmit data in BT-MBAN. In the beginning, the related analysis showed the actual bit rates of MBANs were far below the capacity of a Bluetooth network and long data delays in MBANs can be tolerated. Using Hold mode repeatedly had more power saving and was more flexible than Sniff mode, of course, at the expense of the hold parameters having to be renegotiated.

Next, the core design was developed: Bluetooth devices in a BT-MBAN buffer data and enter Hold mode with a new estimated Hold time $t_h$ and transmit data over an estimated active time $t_a$ and enter hold mode again with an estimated Hold time $t_h$. The Hold time $t_h$ and the active time $t_a$ were determined by the application data profiles and can be the same value when stable. The analysis of specific operations between Bluetooth devices was discussed.

Moreover, the BT-MBAN systems can be considered as a M/G(M/M)/1/N queueing model. The system utilization factor $\rho$ of queueing model reflected the BT-MBAN system performance.

The performance analysis showed when the BT-MBAN system utilization $\rho$ was increased, the active time $t_a$ was reduced while the Hold time is unchanged. With the increasing of BT-MBAN system utilization $\rho$, the power consumption saving utilization $\rho_{saving}$ was increased and more energy was saved. For the same BT-MBAN system utilization $\rho$, the higher the data transmission rate $\mu_a$ at active time, the more energy can be saved.

Finally, in order to reduce the whole power consumption of BT-MBAN system by optimizing their parameters, the queueing model was used to explore and improve the system. The parameters for optimization was the buffer size $N$ and the Hold time $t_h$ in a given BT-MBAN system. Though the analysis and calculation, it was shown that there can be an optimization $t_h$, where the whole power consumption $F$ reached the minimum.
6.6. *Summary and further discussion*

Realistically, there are some issues that should be considered when using the new design.

1. The analysis above has been for a point to point BT-MBAN. When there is more than one slave, the master can poll another slave during the Hold time. Therefore, the master shall manage the network so that the slave always has minimum operations and lower power consumption in the BT-MBAN.

2. Based on the current requirements of an MBAN, the system utilization $\rho$ must be selected to guarantee system round trip, including system stability and reliability.

3. Although the larger Hold time and active time can reduce the amount of times in Hold mode per unit of time, it shall increase the data delay of application. Hence, the designer should take account of other things so that the BT-MBAN can be overall optimized.
Chapter 7

Conclusions & Recommendations

In this dissertation, several new power saving optimization approaches were investigated by the research design, modelling and model analysis, and evaluation in the Bluetooth Wireless Personal Area Networks (WPANs). This chapter summaries the results and contributions arising from the research and suggests directions for future research.

The chapter is structured as follows. Section 7.1 presents the research work summaries for each chapter and Section 7.2 presents the discussion about the novelty and limitations of the research, and makes recommendations for future work.

7.1 Research work summaries

The focus of the research was on the proven Bluetooth BR/EDR technology and the main research objectives, direction and approaches were to investigate low-power operations in Bluetooth BR/EDR technology, with a view to design new optimization approaches of power saving for use with Bluetooth BR/EDR technology. The research work can be summarized as follows.

7.1.1 Research work in Chapter 2

To begin with, Chapter 2 introduced and analyzed the current low-power operations in the Bluetooth BR/EDR controller. The results of the research work related to Chapter 2 are summarized and detailed as follows:

1. The current low power operations, the default polling system and the three
7.1. Research work summaries

low power modes—Sniff, Hold and Park, can reduce the power consumption of Bluetooth.

2. The analysis showed that the designers needed to set the low-power operations and their parameters according to the particular application.

3. The parameters of cross-layer negotiation between the hosts and the controllers often can’t meet the requirements of the variety of applications.

As a result, the research focused on Bluetooth BR/EDR technology and on improving power consumption by the design of new operation modes or approaches. With full analysis, several optimized designs were proposed in Chapters 3, 4 and 6.

7.1.2 Research work in Chapter 3

Chapter 3 was the first optimized design. A Packet Reassembly and Segmentation Protocol (P-RASP) in the Bluetooth baseband was proposed to operate during the idle/sleep interval duration in Bluetooth controllers. The results of the research work related to Chapter 3 are summarized and detailed as follows:

1. The investigation showed: (1) when the rate was quite low and much less than the capacity of Bluetooth, the BR/EDR controllers had to do many polling operations to maintain channel synchronization and deliver the data using small size baseband packet types; (2) many low power operations resulted in the Bluetooth devices having an idle/sleep interval duration.

2. The P-RASP, which can run at idle/sleep interval duration, was designed to reassemble the buffered small size HCI ACL packets to larger ones, so that the Bluetooth LM can assemble a larger baseband packet type (e.g. 3-DH5) with full payload. Using the additional identifier—LLID code ‘00’ in the baseband packets payload header, the P-RASP shall allow the peer Bluetooth device to identify and segment the reassembled packet to restore the original packets.

3. The P-RASP feature can be negotiated by a new LM command or by using the ATT and GATT, and implemented in the Bluetooth BR/EDR controller. Relative to the HCI and the L2CAP layer, the P-RASP was transparent during packet exchange. By the simulation of several scenarios,
the performance analysis revealed that this protocol can significantly reduce the active slots and enhance the efficiency of packet transmission with lower overhead.

4. The usage of P-RASP can be adopted in conjunction with these low power techniques to develop even more efficient packet transmission in the future.

7.1.3 Research work in Chapter 4 and 5

The second optimized design was presented in Chapters 4 and 5. The research proposed a new strategy for reducing power consumption by improving the polling operation. A system model of the proposed state-transition approach was established by the Hidden Markov model (HMM). The results of the research work related to Chapters 4 and 5 are summarized and detailed as follows:

1. The proposed state-transition approach in Chapter 4 was a polling scheduling scheme and can adaptively improve the polling operation in Bluetooth.

2. The new approach used a set of three different polling intervals in the Bluetooth BR/EDR controllers, whereby the controllers can adaptively choose the intervals and link state transfers from active to idle based on a common algorithm. A set of default parameters can be used, and it can easily be added to Bluetooth as a feature. If necessary, additional configuration options can be supported through new HCI and/or LMP commands.

3. The simulation results showed that although the power consumption of the proposed approach was slightly more than adopting Sniff mode for a typical configuration, it had low average end-to-end packet delay and was easier and more flexible for setting the parameters. The state-transition approach will adjust the states according to the traffic levels. The state setting can be optimized for a specific Bluetooth scenario.

4. The established system model in Chapter 5, which was a HMM, can estimate state-transition paths on the controllers and calculate the average power of proposed approach for a generic example. Furthermore, it can also be used to design new state-transition rules for special applications.

5. The HMM can be a common model to analyze the state-transition issues and be used to design and develop more efficient low power modes in Bluetooth.
7.2 Discussion

7.1.4 Research work in Chapter 6

Chapter 6, as the last optimized design, employed a M/G(M/M)/1/N queueing model and proposed a cross-layer approach to transmit data with the Hold mode. The target applications were low rate Bluetooth-based Medical Body Area Networks. The results of the research work related to Chapter 6 are summarized and detailed as follows:

1. A controlled Hold mode can be repeatedly used to transmit data in BT-MBAN, where the related analysis showed the actual bit rates were far below the capacity of a Bluetooth network and long data delays can be tolerated.

2. The BT-MBAN systems can be considered as a M/G(M/M)/1/N queueing model. The system utilization factor $\rho$ of queueing model can reflect the BT-MBAN system performance.

3. The performance analysis showed when the BT-MBAN system utilization $\rho$ was increased, the active time $t_a$ was reduced while the hold time is unchanged.

4. In order to reduce the whole power consumption of a BT-MBAN system, the queueing model can be used to explore and improve the system by optimizing the parameters.

5. Although the larger hold time and active time reduce the amount of times entered into Hold mode per unit of time, it shall increase the data delay of application.

Finally, by using the optimized designs above, the power consumption of Bluetooth BR/EDR technology can be significantly reduced and other network performances in most cases are maintained the same or within an acceptable range. The results of this research may aid research and development teams to model, analyze and design a new operation mode to optimize power saving and improve efficiency for special Bluetooth applications.

7.2 Discussion

The optimized designs in the dissertation are independent of each other but can also be used together. This section will discuss the advantage (novelty) and
7.2. Discussion

disadvantage (limitations) of the research contributions, and make recommendations for future work.

7.2.1 Advantage of research contributions

The guiding ideology of the research, which has been presented in the first paragraph of Chapter 2, is that:

- At the microscopic level, the operations of packet handling and slot occupancy must be minimized. The basic idea is to reduce information exchange between the Bluetooth devices, and allow the transmitter and receiver to return to sleep if possible.

- At the macroscopic level, the basic idea which realizes low power is to adopt low power operations that reduce the duty cycle of the Bluetooth devices.

In a word, the research was trying to achieve power saving for use with Bluetooth BR/EDR technology by improving transmission efficiency, reducing the operations and entering sleep when appropriate.

A. Transmission efficiency improvement

The proposed P-RASP in Chapter 3 can improve the packet transmission efficiency by reducing the active time of the data packet’s delivery in Bluetooth BR/EDR controller.

The proposed state-transition approach in Chapters 4 and 5 used a set of three different polling intervals in the Bluetooth BR/EDR controllers, whereby the controllers can adaptively choose the intervals and link state transfers from active to idle based on a common algorithm. The unnecessary polling operations can be reduced by the state-transition approach. Thus, the link transmission efficiency was improved.

In Chapter 6, a device in low data rate BT-MBAN can buffer data and enter HOLD mode with a new estimated HOLD time $t_h$ and transmit data over an estimated active time $t_a$ and enter hold mode again with an estimated HOLD time $t_h$. Hence, the device’s active time $t_a$ was reduced by entering the HOLD time $t_h$ repeatedly and the transmission efficiency was improved.
7.2. Discussion

B. Reducing the polling operations and entering sleep when appropriate

In Chapters 4 and 5, the kernel idea of a state-transition approach was that the Bluetooth BR/EDR controllers can reduce unnecessary polling operations and enter an idle/sleep state for power saving. In addition, the simulation showed that the state-transition approach had very low average end-to-end packet delay and can adjust the states according to the traffic levels. Therefore, the controllers must be enter sleep with appropriate.

In Chapter 6, based on the typical applications of a BT-MBAN, the repeated use of a controlled Hold mode to transmit data can reduce the active time in a link and the Bluetooth BR/EDR controller can properly enter sleep by this cross-layer approach.

C. The combination of research designs

The P-RASP can be transparent during packet exchange. Once the controllers are aware that their peer supports it in the default scenario, the P-RASP can be combined with other approaches or protocols without additional design.

The operating conditions of the P-RASP protocol in Chapter 3 was the idle/sleep interval duration. Hence, it can be combined with the three polling interval of the proposed state-transition approach in Chapters 4 and 5 and the Hold time $t_h$ duration of the cross-layer approach in Chapter 6.

Moreover, the setting of the three or more polling intervals of the proposed state-transition approach by a HMM can be used in conjunction with the P-RASP and special applications to develop even more efficient packet transmission.

In addition, the P-RASP and the proposed state-transition approach also are applicable during repeated use of a controlled Hold mode to transmit data. The parameter optimization by the queueing model can be used to further explore and improve the whole performance of Bluetooth device or system.

D. The comparison of the research designs to current Bluetooth technology

The P-RASP in Chapter 3 was designed to reassemble the buffered small size HCI ACL packets to larger ones, so that the Bluetooth LM can assemble a larger baseband packet type (e.g. 3-DH5) with full payload. The current Bluetooth mainly uses 1 slot baseband packets (e.g. DM1, DH1, 2-DH1 and 3-DH1) to
7.2. Discussion

deliver data when the applications have a low data rate.

The proposed state-transition approach in Chapter 4 improved on both the existing Bluetooth polling system and Sniff mode, with less active slots and shorter packet delay, resulting in significant advantage. The mathematical modelling in Chapter 5 can be used to design and develop more efficient low power modes in Bluetooth for current BR/EDR technology.

This research work in Chapter 6 extended the use of the hold mode in current Bluetooth technology for a typical application—BT-MBAN. The repeated use of Hold mode has more power saving and flexibility than Sniff mode, of course, at the expense of the hold parameters having to be continuously renegotiated.

7.2.2 Disadvantage of research contributions

The disadvantage of research contributions are as follows.

The P-RASP must run in a low bit error rate channel environment. If the interference of the wireless channel is serious, the small size packets are more advantageous rather than the large size baseband packets.

The simulation scenarios in Chapter 4 are based on random sequences with defined throughput traffic rates and variable data packet intervals. Many applications are much more deterministic with further opportunity for overall optimization and tradeoffs between power saving and average or worst-case end-to-end delay. Detailed channel modelling was not required to compare the proposed approach with the polling system and the Sniff mode, as only the packet pairs between the controllers needed to be considered. In particular, the algorithm handled errors in the transmission by staying in the active state, hence not justifying bit or packet level error processing. However, more accurate power estimations and comparison can be made by using lower level simulation, e.g. Blueware [52] or UCBT [53].

The repeated use of a controlled Hold mode, the devices have to re-negotiate the hold parameters using HCI and LMP commands, which is a disadvantage, especially when low end-to-end delay is required.

7.2.3 Recommendations for future work

The results presented in this thesis can be extended to address the fundamental challenge of power saving in Bluetooth BR/EDR technology. There are a number
of suggestions for further study. These are as follows:

- The P-RASP and the repeated use of a controlled Hold mode must focus on some target applications, e.g. sensor networks, to perform further assessment; the power saving performance analysis needs be further extended from a point to point network to piconet and scatternet.

- The reinforcement learning technology can be developed to predict the polling interval as the common algorithm in the peer controllers. Combined with the HMM, this can be used for developing an intelligent common algorithm or polling scheduling scheme in Bluetooth, which might further improve energy efficiency.

- Further bench studies are required to validate the approaches when adopted within Bluetooth modules.
Bibliography


112


Appendix A

An observation sequence generation with specified probability

An observation sequence in Chapter 4, which is a simulation scenario, is a set of random numbers “1” and “0”, and the probabilities of numbers “1” and “0” are specified. The MATLAB function \textit{rand}() can be used to produce the observation sequences.

First of all, the function \textit{rand}() can produce uniformly distributed pseudo random numbers between 0 and 1. The function \textit{rand}(n) returns an n-by-n matrix containing pseudo random values drawn from the standard uniform distribution on the open interval (0,1) and the function \textit{rand}(m,n) or \textit{rand}([m,n]) returns an m-by-n matrix. Therefore, we can use \textit{rand}(1,length) to produce 1-by-length matrix.

Secondly, initially set the $p$ is the “0” in observation sequence. If the values in the produced 1-by-length matrix are greater than $p$, we set the value as “1”. In contrast, if the values are smaller or equal to than $p$, we set the value as “0”.

Finally, we record the 1-by-length matrix as an observation sequence. The MATLAB code for an observation sequence generation with an specified probability is as follows:

\begin{verbatim}
len = 1000000;  \%Set the observation sequence length;
r = 0.1;        \%Set the "1" probability p in the sequence;
p = 1-r;        \%Set the "0" probability r in the sequence;
seq = (rand(1,len)); \%Produce random numbers between 0 and 1;
seq(find(seq>p)) = 1; \%"1" with specified probability r;
\end{verbatim}
seq(find(seq\leq p)) = 0; "%0" with specified probability p;
dlmwrite('C:\data0_1.txt', seq); %Record simulation scenario.
Appendix B

The simulation code of the proposed state-transition approach in Chapter 4

When the Bluetooth controller runs the algorithm of the proposed state-transition approach in Chapter 4 under a special observation sequence, the following simulation results must be recorded: (1) the sum of the operations, (2) the average packet delay, (3) the total of power consumption and (4) the number of occupied slots of each state in the controller. Changing the observation sequence with different specified probabilities, we can get the different simulation results of the proposed approach.

The MATLAB code for a simulation of the proposed approach in Chapter 4 is as follows:

```matlab
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
* Copyright (c) 2012, University of Limerick, Ireland.
* All rights reserved.
* This is the simulation code of Proposed state-transition approach.
* Version 1.0
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Set the observation sequence length.
len=1000000;
%
%get one of pre-generated observation sequence.
data=dlmread('C:\data0.1.txt');
%
%Initialization of intervals.
small=4;
medium=19;
large=400;
%
```
% Initialization of power consumption.
powerconsumption=0;
Pdata=1;
Ppollnull=0.8;
Pidle=0.01;
%
% Set counter
i=1; % counter
o=0; % calculate the number of operations
state=0; % state counter, 0 to 3 is q1 to q4
Packetdelay=0; % calculate the number of packets delay
%
% Initialization of the number of occupied slots of each state.
q1=0;
q2=0;
q3=0;
q4=0;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Start algorithm %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
while(i<=len)
    if data(i)==1 % data packets
        i=i+1;
o=o+1; % Operation +1 cause by data packets
        powerconsumption = powerconsumption+Pdata*2;
        q1=q1+1; % Data packets processes on State 1
        state=0; % active state
    else % POLL-NULL packets
        state=state+1;
i=i+1;
o=o+1; % Operation +1 cause by POLL-NULL packets
        powerconsumption = powerconsumption+Ppollnull*2;
        q1=q1+1; % POLL-NULL processes on State 1
        if state==1
            i=i+small;
            j=i-small-1; % Temporary counter i for cached data packets
            powerconsumption = powerconsumption+Pidle*2*small;
            q2=q2+1; % on State 2
            % calculate data packets during small interval
            while j<i-1&&j<len
                j=j+1;
                if data(j)==1
                    o=o+1; % Operation +1 cause by cached data packets
                    powerconsumption = powerconsumption+Pdata*2;
                    IV

IV
q1=q1+1;
state=0;  %have data and clear state.
Packetdelay=Packetdelay+(i-j);
end
end
elseif state==2
i=i+medium;
j=i-medium-1;  %Temporary counter i for cached data packets
powerconsumption = powerconsumption+Pidle*2*medium;
q3=q3+1;    %on State 3
%calculate data packets during medium interval
while j<i-1&&j<len
  j=j+1;
  if data(j)==1
    o=o+1;  %Operation +1 cause by cached data packets
    powerconsumption = powerconsumption+Pdata*2;
    q1=q1+1;
    state=0;
    Packetdelay=Packetdelay+(i-j);
  end
end
elseif state==3    %counter for POLL-NULL twice
  %continues active state
elseif state==4
i=i+large;
j=i-large-1;  %Temporary counter i for cached data packets
powerconsumption = powerconsumption+Pidle*2*large;
q4=q4+1;    %on State 4
%calculate data packets during large interval
while j<i-1&&j<len
  j=j+1;
  if data(j)==1
    o=o+1;  %Operation +1 cause by cached data packets
    powerconsumption = powerconsumption+Pdata*2;
    q1=q1+1;
    state=0;
    Packetdelay=Packetdelay+(i-j);
  end
end
state = 2;
else
  %nothing
end
end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% End simulation
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Print the simulation result.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% the sum of operations are:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
operations_of_myapproach=o

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% the average packet delay (ms) is: 
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tdelay_of_myapproach=(Packetdelay/len)*0.625

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% the total of power consumption is: 
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
powerconsumption = powerconsumption/(2*len)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Number of occupied slots of each state: 
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
state1=q1
state2=q2*small
state3=q3*medium
state4=q4*large

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

VI
Appendix C

Publications

The papers, which have been published during my Ph.D. study at the University of Limerick, are as follows.

Conference Papers


- Jiangchuan Wen and John Nelson. Hold Mode for Cross-layer Approach to Transmit Data in Bluetooth-based Body Area Networks. 6th International Mobile Multimedia Communications Conference (MobiMedia), September 2010.


Journal Papers
