



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ ΚΑΙ
ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

ΔΙΑΧΕΙΡΙΣΗ ΣΥΣΚΕΥΩΝ ΣΕ ΔΙΚΤΥΑ
BLUETOOTH

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΧΡΙΣΤΙΝΑ Α. ΚΟΣΣΥΒΑ

Επιβλέπων: Φίλιππος Κωνσταντίνου
Καθηγητής Ε.Μ.Π.

Αθήνα, Νοέμβριος 2009



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ
ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ ΚΑΙ
ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

ΔΙΑΧΕΙΡΙΣΗ ΣΥΣΚΕΥΩΝ ΣΕ ΔΙΚΤΥΑ BLUETOOTH

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΧΡΙΣΤΙΝΑ Α. ΚΟΣΣΥΒΑ

Επιβλέπων: Φίλιππος Κωνσταντίνου
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 13^η Νοεμβρίου 2009.

(Υπογραφή)

.....
Φίλιππος Κωνσταντίνου
Καθηγητής Ε.Μ.Π.

(Υπογραφή)

.....
Αθανάσιος Παναγόπουλος
Λέκτορας Ε.Μ.Π.

(Υπογραφή)

.....
Μιχαήλ Θεολόγου
Καθηγητής Ε.Μ.Π.

Αθήνα, Νοέμβριος 2009

(Υπογραφή)

.....
ΧΡΙΣΤΙΝΑ Α. ΚΟΣΣΥΒΑ

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

© 2009 – All rights reserved

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

ΠΕΡΙΛΗΨΗ

Το Bluetooth είναι μια ασύρματη τεχνολογία χαμηλής ισχύος που επιτρέπει την μικρού εύρους επικοινωνία ανάμεσα σε πολλές συσκευές, και σήμερα αποτελεί έναν παγκόσμιο βιομηχανικό ορισμό για τέτοιου είδους ασύρματη επικοινωνία. Η τεχνολογία του Bluetooth βρίσκει πολλές εφαρμογές στα πεδία του μάρκετινγκ και της διαφήμισης. Το ενδιαφέρον αυτής της τεχνολογίας, στο οποίο οφείλει και την ευρεία εφαρμογή της, είναι το γεγονός ότι επιτρέπει το σχεδιασμό *κυμάτων επικοινωνίας* χαμηλής ισχύος, μικρού μεγέθους και χαμηλού κόστους, τα οποία μπορούν να προσαρμοστούν σε πολλές συσκευές χειρός.

Στην εργασία αυτή θα αναφερθούμε στην προσπάθειά μας να εξερευνήσουμε τη συγκεκριμένη περιοχή εφαρμογών του Bluetooth. Θα παρουσιάσουμε μια εμπειρική έρευνα για την εφαρμογή της τεχνολογίας Bluetooth σε δραστηριότητες του μάρκετινγκ και της διαφήμισης, από τη σκοπιά του χρήστη. Ο βασικός ερευνητικός στόχος αυτής της εργασίας είναι η πειραματική εξέταση της προσοχής που δίνει ο χρήστης στο εξωτερικό του περιβάλλον κατά τη διάρκεια μιας διαδραστικής επικοινωνίας μέσω Bluetooth με το κινητό του τηλέφωνο, π.χ. τι συμβαίνει όταν ο χρήστης λαμβάνει ένα διαφημιστικό μήνυμα μέσω Bluetooth στο κινητό του ενώ περιμένει το λεωφορείο ή οδηγεί το αυτοκίνητό του. Μέσα από μια πειραματική διαδικασία μελετώνται ο χρόνος ανταπόκρισης του χρήστη στο λαμβανόμενο μήνυμα, τα εμπόδια που αντιμετωπίζει από το εξωτερικό περιβάλλον, και γενικότερα ο τρόπος με τον οποίο το προωθούμενο περιεχόμενο επηρεάζει την εμπειρία του χρήστη κατά τη διάρκεια μιας διαδραστικής διαδικασίας μέσω Bluetooth.

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ

Bluetooth, διαδραστική επικοινωνία, χρόνος scan, χρόνος push, εφαρμογή στο κινητό, εφαρμογή στον υπολογιστή, χρόνος απόκρισης στο κινητό, βαθμοί ποιικής στο κινητό, βαθμοί ποιικής στον υπολογιστή.

ABSTRACT

Bluetooth is a low power wireless standard for allowing a short range communications among multiple devices. It has become a global specification for such wireless connectivity. The technology of Bluetooth finds numerous applications in marketing and advertising fields. What makes this technology interesting and wide applicable is the fact that it is wide spread and allows the design of low power, small sized, and low cost radios that could be fit in many handheld devices.

In this paper we introduce our current work on exploring this area. We present an empirical investigation of user experience with Bluetooth-Based marketing and advertising activities. The main research goal of this study is to examine experimentally the attention of the user to his external environment, during an interactive communication by bluetooth with his mobile phone, e.g. what happens when a user receives an advertising message by Bluetooth to his mobile phone while waiting for the bus or driving his car. Response time of the user to the received message or to the obstacles he faces to his external environment and how in general does a pushed content affect the user experience during an interaction via Bluetooth are investigated through an experimental procedure.

KEY WORDS

Bluetooth, interaction via Bluetooth, scan time, push time, mobile application, pc application, response time mobile, penalty points mobile, penalty points pc.

ΕΥΧΑΡΙΣΤΙΕΣ

Ευχαριστώ θερμά τον Καθηγητή και Επιβλέποντα της παρούσας διπλωματικής εργασίας κ. Φ. Κωνσταντίνου, που μου έδωσε την ευκαιρία να ασχοληθώ με ένα τόσο σύγχρονο και ενδιαφέρον αντικείμενο.

Ένα μεγάλο ευχαριστώ επίσης στον Δρ. Δ. Κονταρίνη, στέλεχος της Velti, για την πολύτιμη βοήθεια και καθοδήγησή του κατά τη διάρκεια ανάπτυξης και συγγραφής αυτής της διπλωματικής.

Ιδιαίτερα θέλω να εκφράσω τις ευχαριστίες μου σε όλους τους συμμετέχοντες στην πειραματική διαδικασία που διεξήχθη για την πολύτιμη βοήθειά τους και την εθελοντική τους συμβολή στην πραγματοποίηση του πειραματικού σχεδιασμού της παρούσας διπλωματικής εργασίας.

Ευχαριστώ πολύ την οικογένειά μου και τους φίλους μου για την αμέριστη συμπαράσταση τους καθ' όλη τη διάρκεια των σπουδών μου.

CONTENTS

1. INTRODUCTION.....	15
1.1 About Bluetooth.....	15
1.2 Bluetooth SIG.....	17
1.3 History of Bluetooth Technology.....	18
2. THE TECHNOLOGY OF BLUETOOTH.....	22
2.1 Introduction.....	22
2.2 The Bluetooth Protocol Stack.....	22
2.2.1 The OSI Reference Model.....	24
2.2.2 Masters, Slaves, Slots and Frequency Hopping.....	26
2.2.3 Piconets and Scatternets.....	27
2.2.4 Radio Power Classes.....	28
2.2.5 Voice and Data Links.....	30
2.3 Using Bluetooth.....	31
2.3.1 Discovering Bluetooth Devices.....	31
2.3.2 Connecting to a Service Discovery Database.....	32
2.3.3 Connecting to a Bluetooth Service.....	34
2.3.4 Discoverability and Connectivity Modes.....	35
2.4 Health Concerns.....	36
3. ANTENNAS.....	38
3.1 Introduction.....	38
3.2 Radiation Pattern.....	38
3.3 Gains and Losses.....	40
3.4 Types of Antennas.....	41
3.4.1 Dipole Antennas.....	41
3.4.2 Flat Panel.....	42
3.4.3 Microstrip.....	43
3.5 Ceramic Antennas.....	43
3.6 On-chip Antennas.....	44
3.7 Antenna Placement.....	45

4. ENCRYPTION AND SECURITY.....	46
4.1 Introduction.....	46
4.2 Key Generation and the Encryption Engine.....	48
4.2.1 Encryption Keys.....	49
4.2.2 The E Algorithms.....	51
4.2.3 Key Generation and SAFER+.....	52
4.2.3.1 Ar and A'r.....	53
4.2.3.2 Advantages of SAFER+ and Associated Implementation Issues.....	53
4.3 Secret Keys and Pins.....	54
4.3.1 The Bluetooth Passkey.....	55
4.4 Pairing and Bonding.....	55
4.4.1 Authentication.....	56
4.4.2 Unit Keys.....	58
4.4.3 Link Key Generation.....	59
4.4.4 Changing Link Keys.....	60
4.4.5 Changing to Temporary Link Keys.....	61
4.4.6 Reverting to Semipermanent Link Keys.....	63
4.4.7 Storing Link Keys.....	64
4.4.8 General and Dedicated Bonding.....	65
4.5 Starting Encryption.....	67
4.5.1 Negotiating Encryption Mode.....	67
4.5.2 Negotiating Key Size.....	68
4.5.3 Starting Encryption.....	69
4.5.4 Stopping Encryption.....	69
4.6 Security Modes.....	70
4.7 Security Architecture.....	71
4.7.1 Security Levels.....	71
4.7.2 The Security Manager.....	72
4.7.3 Setting up Security on New Connections.....	74
5. APPLICATIONS.....	76
5.1 Introduction.....	76

5.2 Personal Computers.....	76
5.3 Mobile Phones.....	77
5.4 Gaming.....	77
5.5 Advertising.....	78
6. FUTURE WORK.....	80
6.1 Ultra-wideband (UWB).....	80
6.2 Applications.....	82
6.3 Bluetooth and Ultra-wideband (UWB).....	93
7. EXPERIMENT.....	97
7.1 Introduction.....	97
7.2 Equipment.....	97
7.2.1 Experimental Protocol Brief Description.....	97
7.2.2 PC Application.....	98
7.2.3 Mobile Application.....	100
7.3 Experimental Procedure.....	103
7.4 Related Work.....	104
7.4.1 Scan Time.....	104
7.4.2 Push Time.....	105
7.5 Scan and Push Times.....	107
7.5.1 Scan Time.....	107
7.5.2 Push Time.....	110
7.6 Measurements.....	111
7.6.1 Response Time Mobile.....	111
7.6.2 Penalty Points Mobile.....	114
7.6.3 Penalty Points PC.....	116
8. STATISTICAL ANALYSIS.....	119
8.1 Analysis of Variance (ANOVA).....	119
8.1.1 Between- and Within-Groups Variation.....	119
8.1.2 Statistical Significance.....	120
8.1.3 F-test.....	120

8.2 Response Time Mobile.....	121
8.2.1 Mobile Application A (simple).....	121
8.2.1.1 Scan and Push Times Graphs.....	121
8.2.1.2 ANOVA.....	122
8.2.1.3 Gender and Age Graphs.....	125
8.2.2 Mobile Application B (complex).....	132
8.2.2.1 Scan and Push Times Graphs.....	132
8.2.2.2 ANOVA.....	133
8.2.2.3 Gender and Age Graphs.....	135
8.3 Penalty Points Mobile.....	142
8.3.1 Mobile Application A (simple).....	142
8.3.1.1 Scan and Push Times Graphs.....	142
8.3.1.2 ANOVA.....	143
8.3.1.3 Gender and Age Graphs.....	145
8.3.2 Mobile Application B (complex).....	152
8.3.2.1 Scan and Push Times Graphs.....	152
8.3.2.2 ANOVA.....	153
8.3.2.3 Gender and Age Graphs.....	155
8.4 Penalty Points PC.....	162
8.4.1 Mobile Application A (simple).....	162
8.4.1.1 Scan and Push Times Graphs.....	162
8.4.1.2 ANOVA.....	163
8.4.1.3 Gender and Age Graphs.....	165
8.4.2 Mobile Application B (complex).....	172
8.4.2.1 Scan and Push Times Graphs.....	172
8.4.2.2 ANOVA.....	173
8.4.2.3 Gender and Age Graphs.....	175
8.5 Total Results.....	182
8.5.1 Mobile Application A (simple).....	182
8.5.2 Mobile Application B (complex).....	182
8.5.3 Conclusions.....	182
9. REFERENCES.....	183

1

INTRODUCTION

1.1 About Bluetooth

Bluetooth is a low-power, short-range wireless technology originally developed for replacing cables when connecting devices like mobile phones, headsets and computers. It has since evolved into a wireless standard for connecting electronic devices to form Personal Area Networks (PANs) as well as ad hoc networks. Not only will cables be unnecessary for connecting devices, but connections will also be done seamlessly without the need for installations and software drivers. With this technology, devices will be able to discover any other Bluetooth-enabled device, determine its capabilities and applications, and establish connections for data exchange.

Bluetooth was named after a late tenth century Danish Viking king, Harald Blatand (Blatand is Danish for Bluetooth), who united and controlled Denmark and Norway. The name was adopted because Bluetooth wireless technology was expected to unify the telecommunications and computing industries.

During Bluetooth's inception, its developers envisioned several usage scenarios. The following examples illustrate five of these:

- *Three-in-one phone - use the same phone everywhere*

This is a wireless phone that will use the best telecommunication technology available. At the office, it will use Bluetooth technology to communicate with other phones thereby acting as an intercom or a walkie-talkie. At home, it will function as a cordless phone, incurring fixed-line charges. When the user is on the move, it can function as a mobile (cellular-like) phone.

- *Internet bridge - surf the Internet regardless of the connection*

The user will be able to connect to the Internet anywhere, regardless of whether it is through a wireless connection using a Bluetooth link with a mobile phone or a wired connection such as a local area network (LAN), a public switched telephone network (PSTN) or a digital subscriber line (DSL).

- *Interactive conference - connect every participant for instant data exchange*

In conferences or meetings, participants will be able to instantly exchange information such as business cards or presentation slides using their Bluetooth-enabled devices.

- *The ultimate headset - a cordless headset keeps your hands free*

This is a Bluetooth-enabled headset that allows users to connect wirelessly to their mobile phones or mobile PCs for a hands-free connection, giving users the flexibility to concentrate on more important matters.

- *Automatic synchronization*

Personal devices such as a desktop computer, hand-held, mobile phone and notebook belonging to the same user will perform automatic synchronization of their Personal Information Management (PIM) applications. When the user enters the office, the calendar application on the user's mobile phone or handheld automatically synchronizes with the scheduler in the office, alerting the user of any conflicts in his schedule or upcoming meetings.

Today, scenarios such as those depicted in [Figure 1.1](#), illustrate a vision for ad hoc networks connected by Bluetooth links. In the figure, devices belonging to one user can interconnect with each other and they can also connect to local information points - in this example, to get updates on flight arrivals and departures.

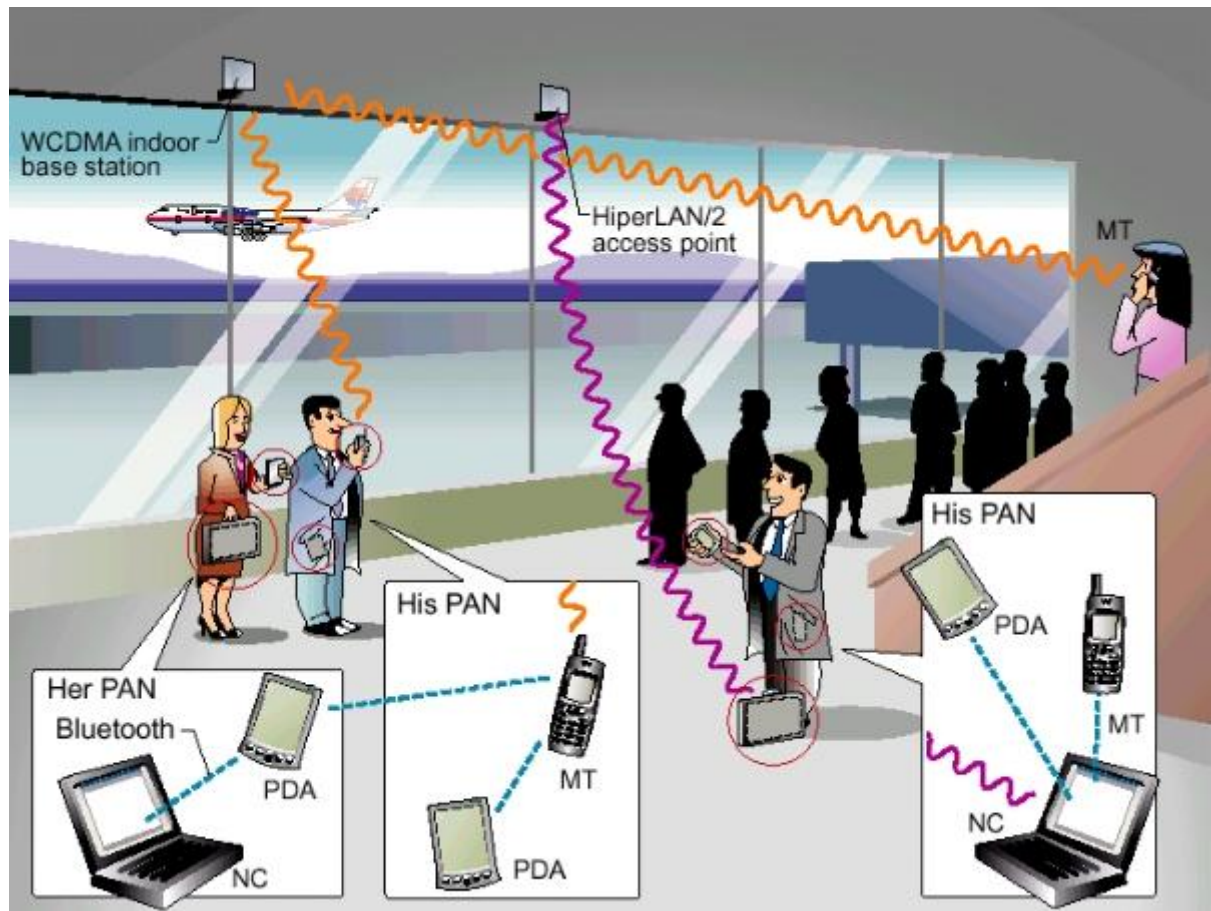


Figure 1.1: Ad hoc networks at an airport scenario.

1.2 Bluetooth SIG

The Bluetooth Special Interest Group (SIG) is a privately held, not-for-profit trade association founded in September 1998. The Bluetooth SIG itself does not make, manufacture, or sell *Bluetooth* enabled products. The SIG member companies are leaders in the telecommunications, computing, automotive, music, apparel, industrial automation, and network industries. SIG members drive development of *Bluetooth* wireless technology, and implement and market the technology in their products. The main tasks for the Bluetooth SIG are to publish *Bluetooth* specifications, administer the qualification program, protect the *Bluetooth* trademarks and evangelize *Bluetooth* wireless technology.

The Bluetooth SIG global headquarters are in Bellevue, Washington, USA and has local offices in Hong Kong, Beijing, China, Seoul, Korea, Minato-Ku, Tokyo, Taiwan and Malmo, Sweden.

In addition to the Bluetooth SIG staff, volunteers from member companies play key roles in furthering *Bluetooth* wireless technology and the organization behind it.

Members support a number of working groups and committees that focus on specific areas, such as engineering, qualification, and marketing.

The Bluetooth SIG includes Promoter member companies Ericsson, Intel, Lenovo, Microsoft, Motorola, Nokia, and Toshiba, and thousands of Associate and Adopter member companies.

1.3 History of Bluetooth technology



- The Bluetooth Special Interest Group (SIG) is formed with five companies.
- The Bluetooth SIG welcomes its 400th member by the end of the year.
- The name *Bluetooth* is officially adopted.



- The *Bluetooth* 1.0 Specification is released.
- The Bluetooth SIG hosts the first UnPlugFest for member engineers.
- *Bluetooth* technology is awarded "Best of Show Technology Award" at COMDEX.



- First mobile phone.
- First PC Card.
- Prototype mouse and laptop demonstrated at CeBIT 2000.
- Prototype USB dongle shown at COMDEX.
- First chip to integrate radio frequency, baseband, microprocessor functions and *Bluetooth* wireless software.
- First Headset.
- First printer.



- First laptop.
- First hands-free car kit.
- First hands-free car kit with speech recognition.
- The Bluetooth SIG, Inc. is formed as a privately held trade association.



- First keyboard and mouse combo.
- First GPS receiver.
- *Bluetooth* wireless qualified products now number 500.
- IEEE approves the 802.15.1 specification to conform with *Bluetooth* wireless technology.
- First digital camera.



- First MP3 player.
- *Bluetooth* Core Specification Version 1.2 adopted by the Bluetooth SIG.
- Shipment of *Bluetooth* enabled products hits rate of 1 million per week.
- First FDA-approved medical system.



- The Bluetooth SIG adopts Core Specification Version 2.0 + Enhanced Data Rate (EDR).
- *Bluetooth* technology reaches an installed base of 250 million devices.
- Product-shipment rate surpasses 3 million per week.
- First stereo headphones.
- Product shipments soar to 5 million chipsets per week.



- The Bluetooth SIG welcomes its 4,000th member.
- The Bluetooth SIG Headquarters opens in Bellevue, WA; regional offices open in Malmo, Sweden and Hong Kong.
- First Sunglasses.



- First watch.
- First picture frame.
- *Bluetooth* wireless reaches an installed base of 1 billion devices.
- *Bluetooth* enabled devices ship at a rate of 10 million per week.
- The Bluetooth SIG announces it will integrate *Bluetooth* technology with the WiMedia Alliance version of UWB.
- First alarm-clock radio.



- First television.
- The Bluetooth SIG welcomes its 8,000th member.
- **SIGNature**, The *Bluetooth* quarterly, makes its debut at the Bluetooth SIG's All Hands Meeting in Vienna, Austria.
- Bluetooth SIG Executive Director, Michael Foley, wins Telematics Leadership Award.



- 2008 marks *Bluetooth* technology's 10 year anniversary - no other wireless technology has grown to be shipping nearly 2 Billion products in 10 years.
- The Bluetooth SIG welcomes its 10,000th member.



- The Bluetooth SIG adopts Core Specification Version 3.0 + HS making *Bluetooth* high speed technology a reality.
- The Bluetooth SIG welcomes its 12,000th member.
- The Bluetooth SIG All Hands Meeting is held in Tokyo—the first AHM in APAC.

2

THE TECHNOLOGY OF BLUETOOTH

2.1 Introduction

Bluetooth is a low-power, short-range wireless technology originally developed for replacing cables when connecting devices like mobile phones, headsets and computers. Bluetooth operates on the unlicensed Industrial Scientific Medical (ISM) band at 2.4 GHz, which ensures worldwide communication compatibility. Since the ISM band is open to anyone, systems operating on this band must deal with several unpredictable sources of interference, such as microwave ovens, baby monitors and 802.11 wireless networks. Hence, to minimize the risk of such interference, Bluetooth uses a Frequency Hopping Spread Spectrum (FHSS) technology for its air interface. During a connection, radio transceivers hop from one channel to another. This means that after one packet is sent on a channel, the two devices retune their frequencies (hop) to send the next packet on a different channel. When the transmission encounters a disturbance due to interference, the packet will simply be retransmitted on a different channel. Hence, if one frequency channel is blocked, there will be a limited disturbance to the Bluetooth communication. This allows several Bluetooth networks to run concurrently without interrupting one other. The link rate offered by Bluetooth is 1 Mbps, but with overhead, this effectively becomes 721 kbps. The typical range for Bluetooth is 10m, but it can reach up to 100m depending on the power class of the device.

2.2 The Bluetooth Protocol Stack

A key feature of the Bluetooth specification is that it aims to allow devices from lots of different manufacturers to work to one another. To this end, Bluetooth does not just define a radio system, it also defines a software stack to enable applications to

find other Bluetooth devices in the area, discover what services they can offer and also use those services.

The Bluetooth stack is defined as a series of layers, though there are some features which cross several layers.

Every block in Figure 2.1 corresponds to a chapter in the core Bluetooth specification. The core specification also has three chapters on test and qualification:

1. Bluetooth test mode.
2. Bluetooth Compliance Requirements.
3. Test Control Interface.

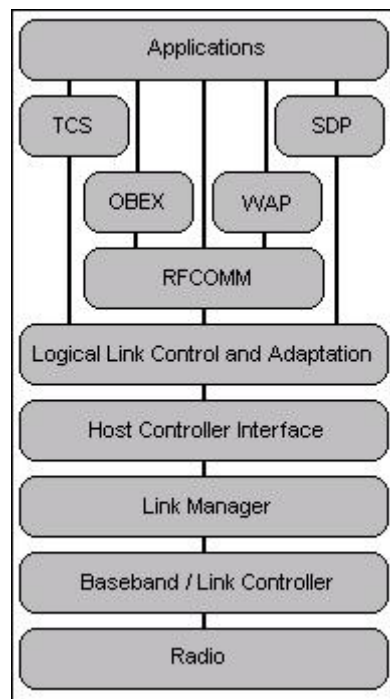


Figure 2.1: The Bluetooth protocol Stack.

- **RADIO**: The radio modulates and demodulates data for transmission and reception on air.
- **BASEBAND/LINK CONTROLLER**: The Baseband and Link Controller control the physical links via the radio, assembling packets and controlling frequency hopping.
- **LINK MANAGER**: The link manager controls and configures links to other devices.

- HOST CONTROLLER INTERFACE: The Host Controller Interface handles communications between a separate host and a Bluetooth module.
- LOGICAL LINK CONTROL AND ADAPTATION PROTOCOL: Logical Link Control and adaptation multiplexes data from higher layers and converts between different packet sizes.
- RFCOMM: RFCOMM provides an RS232 like serial interface.
- WAP AND OBEX : WAP and OBEX provide interfaces to the higher layer parts of the Communications Protocols.
- SDP: SDP lets Bluetooth devices discover what services other Bluetooth devices support.
- TCS: TCS provides telephony services.
- APPLICATIONS: The Bluetooth Profiles give guidelines on how applications should use the Bluetooth protocol stack.

The Bluetooth specification encompasses more than just the core specification. There are also profiles which give details of how applications should use the Bluetooth protocol stack.

2.2.1 The OSI Reference Model

Figure 2.2 shows the familiar Open Systems Interconnect (OSI) standard reference model for communications protocol stacks. Although Bluetooth does not exactly match the model, it is a useful exercise to relate the different parts of the Bluetooth stack to the various parts of the model. Since the reference model is an ideal, well-partitioned stack, the comparison serves to highlight the division of responsibility in the Bluetooth stack.

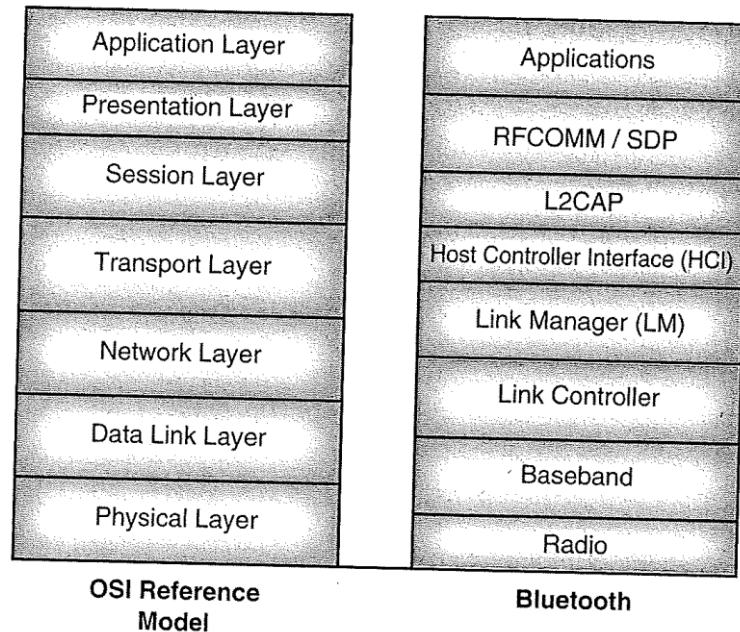


Figure 2.2 : OSI reference model and Bluetooth.

The Physical Layer is responsible for transmission, framing and error control over a particular link and as such overlaps the link controller task and the control end of the baseband, including error checking and correction.

From now on, it gets a little less clear. The Network Layer is responsible for data transfer across the network, independent of the media and specific topology of the network. This encompasses the higher end of the link controller, setting up and maintaining multiple links and also covering most of the Link Manager task. The Transport Layer is responsible for the reliability and multiplexing of data transfer across the network to the level provided by the application and thus overlaps at the high end of the Link Manager and covers the Host Controller Interface (HCI), which provides the actual data transport mechanisms.

The Session Layer provides the management and data flow control services, which are covered by L2CAP and the lower ends of RFCOMM/SDP. The Presentation Level provides a common representation for Application Layer data by adding service structure to the units of data, which is the main task of RFCOMM/SDP. Finally, the Application Layer is responsible for managing communications between host applications.

2.2.2 Masters, Slaves, Slots and Frequency Hopping

Bluetooth devices can operate in two modes: as a master or as a slave. It is the master that sets the frequency hopping sequence. Slaves synchronize to the master in time and frequency by following the master's hopping sequence.

Every Bluetooth device has a unique Bluetooth device address and a Bluetooth clock. The baseband part of the Bluetooth specification describes an algorithm which can calculate a frequency hop sequence from a Bluetooth device address and a Bluetooth clock. When slaves connect to a master, they are told the Bluetooth device address and clock of the master. They then use this to calculate the frequency hop sequence. Because all slaves use the master's clock and address, all are synchronized to the master's frequency hop sequence.

In addition to controlling the frequency hop sequence, the master controls when devices are allowed to transmit. The master allows slaves to transmit by allocating slots for voice traffic or data traffic. In data traffic slots, the slaves are only allowed to transmit when replying to a transmission to them by the master. In voice data slots, slaves are required to transmit regularly slots whether or not they are replying to the master.

Bluetooth



Figure 2.3 : Point to point and point to multipoint piconets.

The master controls how the total available bandwidth is divided among the slaves by deciding when and how often to communicate with each slave. The number of time slots each device gets depends on its data transfer requirements. The system of dividing time slots among multiple devices is called Time Division Multiplexing (TDM).

2.2.3 Piconets και Scatternets

Two or more Bluetooth units sharing the same channel form a piconet. One device acts as a master and the devices connected to it act as slaves. The slaves in a piconet can only have links to the master. Slaves cannot directly transmit data to one another. In effect, the master acts as a switch for the piconet and all traffic must pass through the master. Any device can be either a master or a slave within a piconet and they can change roles at any point in a connection when a slave wants to take over a master's role. There can be up to 7 active slaves in a piconet but only one master.

Every Bluetooth device has its own clock and can be uniquely identified by its Bluetooth device address. Slaves in a piconet use the master's Bluetooth device address and clock to determine the frequency hopping sequence. Offsets are added to the native clocks of each of the slaves to synchronize with the master's clock for the duration of the connection. Furthermore, the master also controls when devices transmit data, since slaves can only transmit when scheduled by the master. Hence, in deciding when and how often it communicates with the slaves, the master effectively controls how the total available bandwidth is distributed among the slaves.

A set of two or more interconnected piconets form scatternets. Figure 2.4 shows an illustration of piconets and scatternets. A Bluetooth unit can be a slave in two or more piconets, but it can be a master in only one. Devices that participate in two or more piconets may act as gateways, forwarding traffic from one piconet to another. Moreover, since Bluetooth units can only transmit and receive data in one piconet, its participation in several piconets is on a Time Division Multiplexing (TDM) basis. This means that although devices can participate in several piconets, they may be active in only one piconet at any one time. Hence, devices participating in multiple piconets divide their time between the piconets, spending some time slots in one and some time slots in another. Piconets may be identified by the master's identity and clock. A device wishing to be active in another piconet will have to notify the master of its current piconet that it will be inactive for a predetermined length of time. The device will then have to re-synchronize its clock (by adding an offset) with its other master. When a slave becomes inactive in a piconet, communications between masters and the other active slaves go on as normal. On the other hand, when a master

becomes inactive in its piconet, the slaves will have to wait for it to be active again before communication can resume.

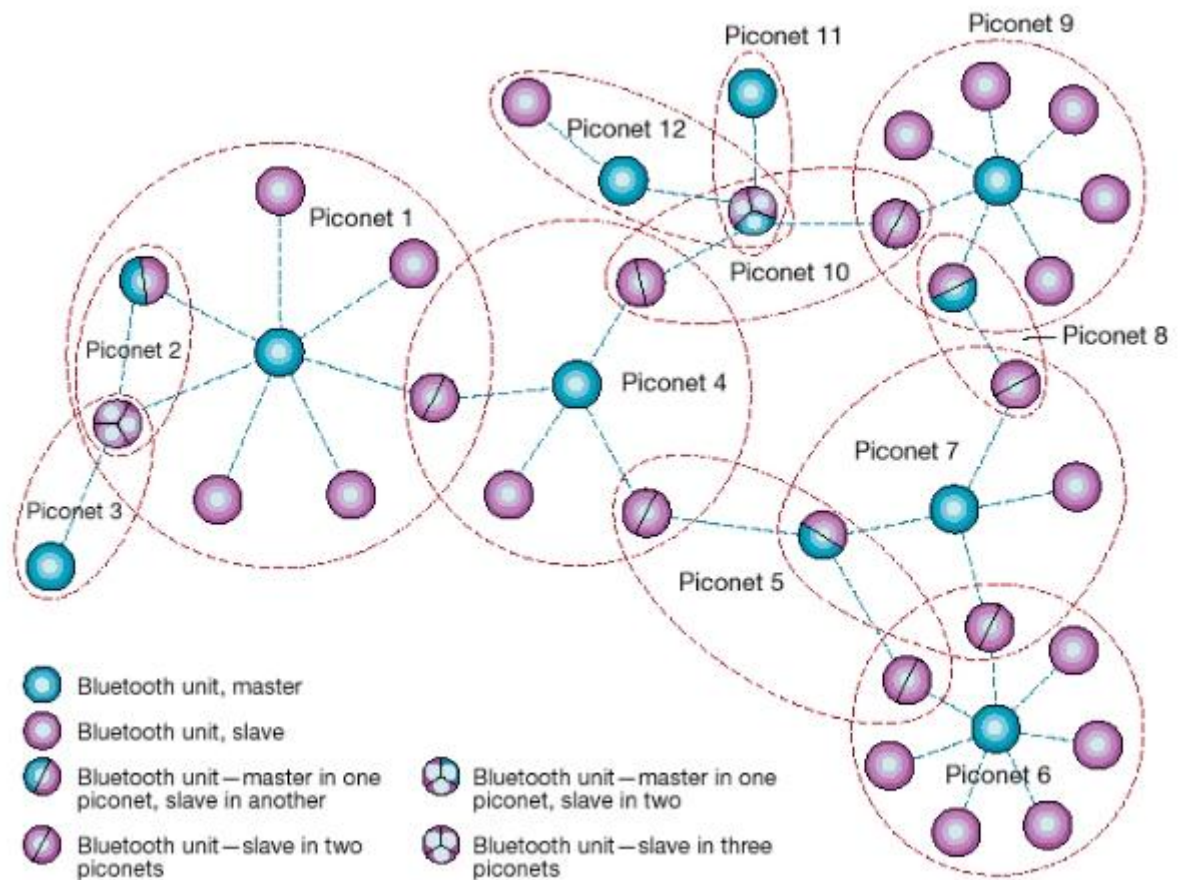


Figure 2.4 : Bluetooth Piconets and Scatternets.

2.2.4 Radio Power Classes

The Bluetooth specification allows for three different types of radio powers:

- Class 1 = 100mW (20dBm)
- Class 2 = 2,5mW (4dBm)
- Class 3 = 1mW (0dBm)

These power classes allow Bluetooth devices to connect at different ranges. At the time of writing, most manufacturers are producing class 3, low power 1mW radios. These can communicate for maximum of around 10m. However, because things like bodies and furniture absorb microwaves, reception may not be reliable at the limit of this range. So, when using 1mW radios, a more realistic figure for reliable operation

in a normal room will probably be 5m. This provides a low cost, low power communications solution which has plenty of range for a cable replacement technology.

Obviously, higher power radios have longer ranges. The maximum range for a class 1, 100mW radio is about 100m. There is also a minimum range for a Bluetooth connection. If radios are put too close together, some receivers may saturate, so a few Bluetooth radios may be unreliable on short link lengths (under 10 cm).

A 100m link needs a high power class 1 device at both ends, but it is possible to create piconets with a mixture of high and low power devices at different ranges. Figure 2.5 shows a mixture of high and low power devices in different piconets occupying an area.

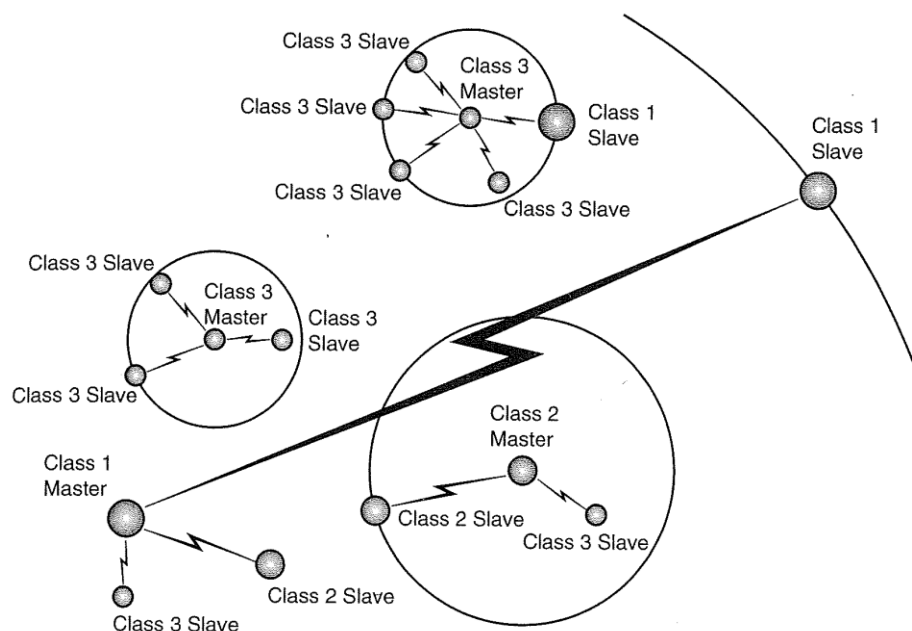


Figure 2.5 : Piconets made up of different power class devices.

This figure shows piconets which overlap each other. This is possible because each master has its own frequency hopping sequence, so two piconets are unlikely to be on the same frequency at the same time. If they do meet on the same frequency, after the next frequency hop they will not still be on the same frequency, so the data which may have been lost can be resent.

2.2.5 Voice and Data Links

Bluetooth allows both time critical data communication such as that required for voice or audio, as well as high speed, time insensitive packet data communication. To carry such data, two different types of links are defined between any two devices. These are SCO (Synchronous Connection Oriented) links for voice communication and ACL (Asynchronous Connectionless) links for data communication.

ACL data packets are constructed from a 72-bit access code, a 54-bit packet header and a 16-bit CRC code, in addition to the payload data. There is a variety of packet types allowing different amounts of data to be sent. The packet which carries the largest data payload is a DH5 packet, which stretches over five slots. A DH5 packet can carry 339 bytes or 2712 bits of data. So 2858 bits are sent on air for 2712 bits of information.

A DH5 packet uses up five slots and the minimum length reply is one slot. Thus, the maximum baseband data rate in one direction is 723,2kb/sec. In this case, with 5-slot packets sent in one direction, the 1-slot packets sent in the other direction will only carry 57, 6kb/sec, so this would be an asymmetric link with more data going in the direction using 5-slot packets. If 5-slot packets were sent in both directions, the data rate obtained would be 433,9kb/sec, quite a reduction from the 1Mb/sec data rate on air.

This overhead in both data encoding and frequency hopping is necessary mainly to provide a robust link since the ISM band is shared resource with many devices and indeed other communications standards and even noise sources, cohabiting in the same spectrum. In addition, to further reduce the interference problem in the spectrum, national radio regulations limit the power emission per unit time in the ISM band, making a frequency hopping scheme necessary to spread transmissions over the spectrum and over time.

The higher layers of the protocol stack also use up some of the bandwidth, so at the application level, the maximum data rate could be around 650kb/sec.

The SCO links work at 64kb/sec and it is possible to have up to three full-duplex voice links at once or to mix voice and data. These voice channels give audio communication of a quality one would expect from a modern mobile cellular phone

system such as GSM. As such, SCO links are not really suitable for delivering audio of a quality required for music listening.

One alternative to support music delivery is to use an ACL channel to carry audio. Raw CD-quality audio requires 1411,2kb/sec, but with suitable compression, such as MP3, which reduces this bit rate to around 128kb/sec, near CD-quality audio could easily be carried providing the time criticality of the audio was maintained.

2.3 Using Bluetooth

Bluetooth is unlike any wired network, as there is no need to physically attach a cable to the devices you are communicating with. Indeed, you may not know exactly what devices you are talking to and what their capabilities are. To cope with this, Bluetooth provides inquiry and paging mechanisms and a Service Discovery Protocol (SDP).

This section examines how these mechanisms are used to allow Bluetooth devices to link up and use one another's services.

2.3.1 Discovering Bluetooth Devices

Imagine two Bluetooth enabled devices, say, a cell phone and a laptop computer. The cell phone is capable of acting as a modem using the dial up networking profile and it periodically scans to see if anyone wants to use it.

The user of the laptop opens up an application that needs Bluetooth dial up networking connection. To use this application, the laptop knows it needs to establish a Bluetooth link to a device supporting the dial up networking profile. The first stage in establishing such a connection is finding out what Bluetooth enabled devices are in the area, so the laptop performs an inquiry to look for devices in the neighborhood.

To do this the laptop transmits a series of inquiry packets and eventually the cell phone replies with a Frequency Hop Synchronization (FHS) packet. The FHS packet contains all the information that the laptop needs to create a connection to the cell phone. It also contains the device class of the cell phone, which consists of major and minor parts. The major device class tells the laptop that it has found a phone. The minor part says that the type of phone is a cellular phone. This exchange of messages is illustrated in Figure 2.6.

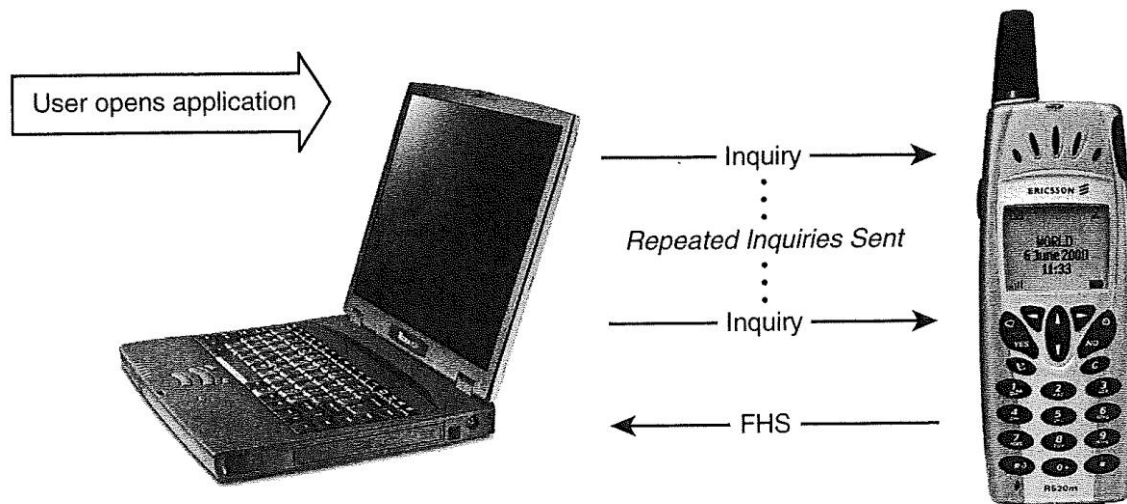


Figure 2.6 : Discovering a Bluetooth device.

In the same way, every Bluetooth enabled device in the area that is scanning for inquiries will respond with an FHS packet, so the laptop accumulates a list of devices.

What happens next is up to the designer of the application. The laptop could present the user with a list of all the devices it has found and let the user choose what to do next. But if it did that at this stage, all it could do is tell the user about the types of devices it has found. Instead of telling the user about the devices it has found, the application could automatically go on to the next stage and find out which devices in the area support the dial up networking profile.

2.3.2 Connecting to a Service Discovery Database

To find out whether a device supports a particular service, the application needs to connect to the device and use the Service Discovery Protocol (SDP). Figure 2.7 shows how this is done. First the laptop pages the cellular phone, using the information it gathered during inquiry. If the phone is scanning for pages, it responds and an ACL baseband connection can be set up to transfer data between the two devices.

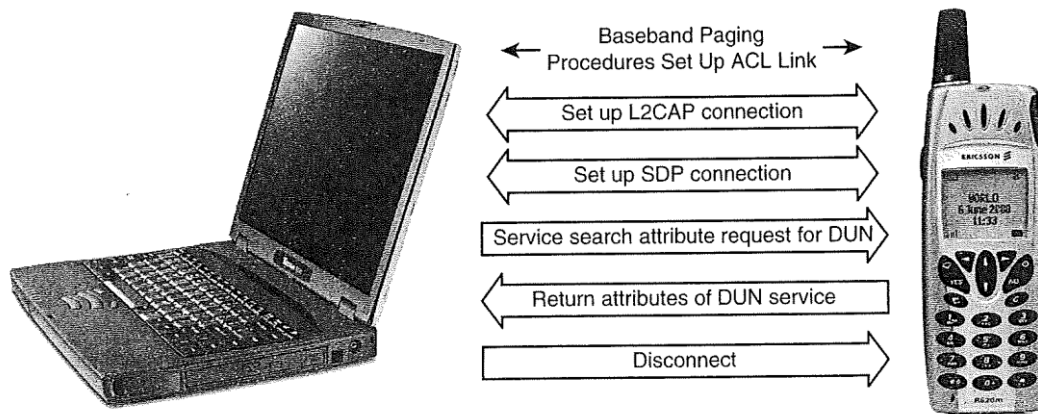


Figure 2.7 : Retrieving information on services.

Once an ACL connection has been established, a Logical Link Control and Adaptation Protocol (L2CAP) connection can be set across it. An L2CAP connection is used whenever data has to be transferred between Bluetooth devices. L2CAP allows many protocols and services to use one baseband ACL link. L2CAP distinguishes between different protocol and services using an ACL connection by adding a Protocol and Service Multiplexor (PSM) to every L2CAP packet. The PSM is different for every protocol or service that uses the link. Since this connection will be used for service discovery, its PSM = 0x0001, a special value that is always used for service discovery.

The laptop uses the L2CAP channel to set up a connection to the service discovery server on the cellular phone. The laptop's service discovery client can then ask the cellular phone's service discovery server to send it all the information it has relating to the dial up networking profile. The service discovery server on the cellular phone searches through its database and returns the attributes relating to the dial up networking.

Once the service discovery information has been received, the laptop may decide to shut down the connection to the cellular phone. If the laptop wants to collect service discovery information from many devices in the area, then it makes sense to shut down the links after using them, since one device can only use a limited number of links at a time, and keeping the links alive will consume battery power unnecessarily.

After the laptop has collected service discovery information from devices in the area, what happens next is again up to the application. It could display the information on all devices it has found which support the dial up networking profile and let the

user decide which one to connect to. Alternatively, the application could decide for itself which device to use without bothering the user.

Either way, the service discovery information tells the laptop everything it needs to know to connect to the dial up networking serviced on the cellular phone.

2.3.3 Connecting to a Bluetooth Service

The process of actually making a connection is shown in Figure 2.8. The paging process which establishes a baseband ACL link is the same as was used when connecting for service discovery.

This time the link is being set up for a protocol which may have particular quality of service requirements, so the application running on the laptop may wish to configure the link to meet its requirements. This is done by the application sending its requirements to the Bluetooth module using the Host Controller Interface. Next, the module's link manager configures the link using the link management protocol.

Once the ACL connection is set up to the laptop's satisfaction, an L2CAP connection is set up. The dial up networking profile uses RFCOMM, an RS-232 emulation layer, so the L2CAP connection uses the Protocol Stack Multiplexor for RFCOMM (PSM = 0x0003).

After the L2CAP link has been set up, an RFCOMM connection can be set up across it. RFCOMM, like L2CAP, can multiplex several protocols or services across one connection. Each protocol or service is given its own channel number. The cellular phone's channel number for dial up networking was sent to the laptop in the service discovery information, so the laptop knows which channel number it should use when setting up the RFCOMM connection.

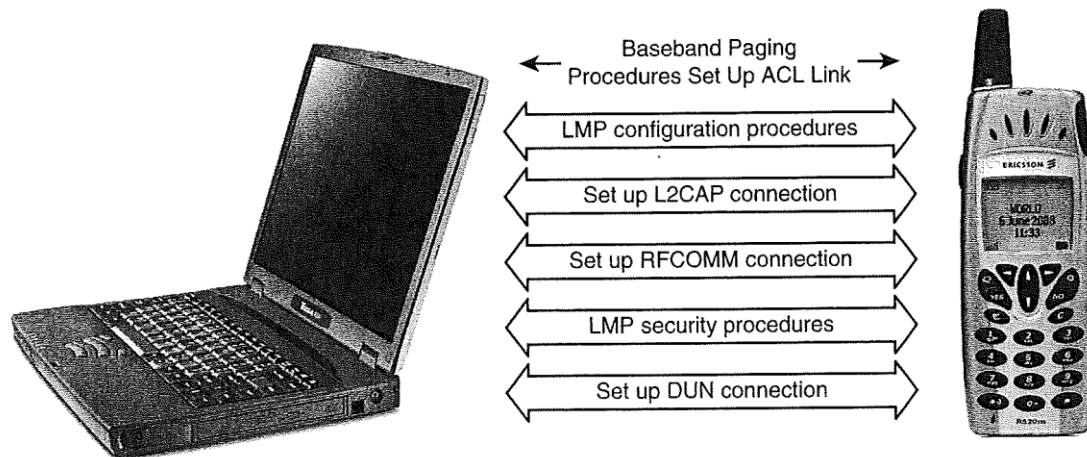


Figure 2.8 : Connecting to a Dial Up Networking service.

Finally, the Dial Up Networking (DUN) connection is set up using the RFCOMM connection, and the laptop can start to use the dial up networking services of the cellular phone. Now, the laptop can use the cellular phone to make connections across the phone network without the two needing to be joined together by a data cable.

If the cellular phone is picked up and taken out of the range of the laptop, the laptop will have to repeat the procedure and find another device to connect to. Meanwhile, the cellular phone is still scanning and might be connected to another device elsewhere. The process of connecting is ad hoc and arbitrary with Bluetooth connections, possibly only lasting for a short period of time as devices move around.

2.3.4 Discoverability and Connectability Modes

It is important to realize that for a connection to be established using Bluetooth wireless technology, both ends of the link have to be willing to connect.

Some devices may be set so that they will not scan for inquiries. In this case, other devices cannot discover them and they will effectively be invisible. Similarly, some devices may be set so that they do not perform page scans. In these cases, they can still initiate connections, but they will not hear other devices trying to connect to them.

Applications can choose whether to make devices connectable or discoverable. A connection cannot be forced on a device which is not in the correct mode to accept it.

2.4 Health Concerns

Bluetooth uses frequency spectrum in the range of 2400 MHz to 2483,5 MHz. This range encompasses the natural frequency of H₂O molecular oscillation at 2450MHz, which is also used by microwave ovens specifically to excite water molecules inside food in order to cook it.

Sharing the same frequency range as microwave ovens has led to some concerns that Bluetooth devices might “cook” their users. Some microwave radiation will be absorbed in flesh. It will be absorbed by field-induced rotation of polarized water molecules, which is converted to heat through molecule friction. Basically, the microwaves shake the water in flesh and it heats up as it shakes. But, as the radiated output power of Bluetooth devices is incredibly low and spread in spectrum in time, experts concur that Bluetooth radiation does not pose a risk to health.

A 1mW Bluetooth radio emits 1/1000000 the amount of power in a 1KW microwave oven. Also, in a microwave oven, all the power is directed inward at the food, whereas in a Bluetooth device, the power is radiated outward, so the user only ever intercepts the smallest fraction of the radio waves which are heading in their direction.

It is interesting to compare Bluetooth devices with other popular communications devices. Bluetooth operates at 2,4 GHz and uses 1mW (0dBm) for most applications, with a maximum of 100 mW (20dBm) for extended range. This means that Bluetooth signals have a penetration depth of only 1,5cm into flesh. In comparison, cellular handsets have a power of 10mW to 2W peak, using 450MHz to 2200MHz and exhibit a penetration depth of 2,5cm in the middle of their range at 900MHz. So, mobile cellular handsets give rise to a measurable heating effect of 0,1 °C, compared with no measurable increase for Bluetooth devices. Although studies have shown this small heating effect, it is too low to be noticed by the user. Most of the temperature increases that mobile users feel when holding a handset next to their ears is caused by an insulating effect. Since the head radiates a lot of heat, if a handset blocks the radiation, then the head heats up. Getting a hot ear from a mobile phone is not necessarily a sign that you are absorbing radiation.

There has already been some controversy regarding cellular handsets and whether they have a negative impact on health. Although scientific opinion is pretty

conclusive that there are no risks, to be safe, various organizations have undertaken studies and research and have laid down guidelines for exposure to radio frequencies.

The WHO, ICNIRP and IEEE have developed Radio Frequency (RF) exposure recommendations and these guidelines have been adopted by many national authorities. In the usual way of health and safety guidelines, they incorporate large safety margins. The guidelines specify near-field restrictions (referred to as SAR) between 10MHz to 10GHz, which devices with an output power of less than 1,6mW are incapable of exceeding. So, all low power Bluetooth devices will fall within these restrictions. Higher power Bluetooth devices may need to be tested for SAR limits, and this will be done as part of radios regulatory testing.

The guidelines also specify a standard for total RF exposure. This is given as a power density of 10 W/m². This level of spectral density would require an unrealistic number of Bluetooth devices to operate continuously in a very small space, which would actually not be possible due to the limited spectrum in the ISM band.

Several expert panels formed from organizations such as WHO, ICNIRP, EC and the Royal Society of Canada have debated the topic of health in the context of existing higher power cellular technology in recent years. They have all concluded that there is no credible or convincing evidence that RF exposure from wireless devices operating within accepted exposure limits causes adverse human health effects. They did, however, recommend additional research to clarify some areas and fill gaps in existing knowledge.

In conclusion, experts agree that Bluetooth devices are too low in power to have any negative health consequences. Even the higher power devices are an order of magnitude lower in power than existing cellular devices, which based on existing research and official guidelines, have already been proven to be safe.

3

ANTENNAS

3.1 Introduction

The antenna transmits and receives the radio waves which Bluetooth wireless technology uses to communicate, so it is a crucial part of any Bluetooth implementation. At their crudest, 2,4GHz antennas can be simple lengths of wire, but for the best performance and to fit well with a product's form factor, more sophisticated antenna designs are required.

When choosing and positioning antennas, the surrounding environment has to be taken into account, as this can have marked effects on antenna performance.

3.2 Radiation Pattern

Antenna radiation patterns are usually plotted in two dimensions: azimuth and elevation. The azimuth pattern is the pattern of radiation looking down on the antenna from above. The elevation pattern is the pattern of radiation looking at the antenna from the side.

A dipole antenna radiates in a torus (a doughnut shape). Figure 3.1 shows the radiation patterns this gives. At the left is the azimuth pattern. The radiation is a perfect circle with the same strength in all directions. At the right is the elevation pattern. At the sides of the antenna, the radiation is strong, dropping away to nothing above and below the antenna.

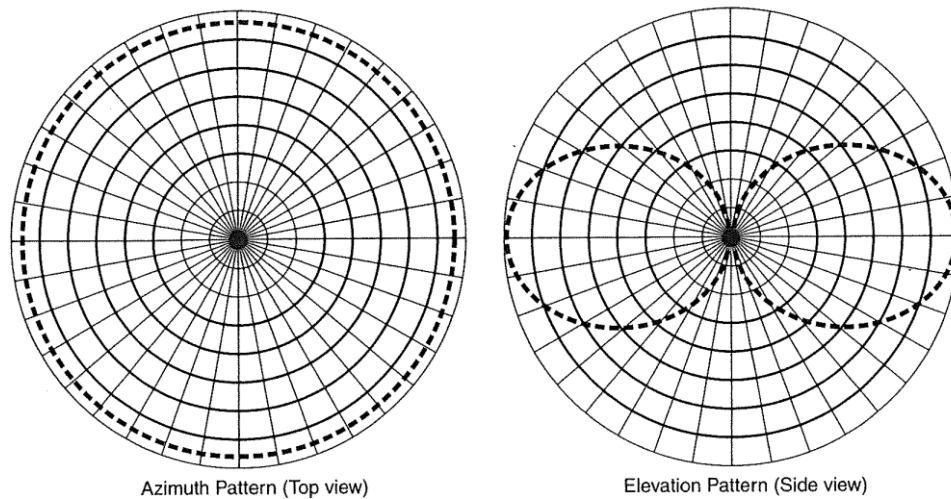


Figure 2-1 Azimuth and elevation patterns for a dipole antenna.

Figure 3.1 : Azimuth and elevation patterns for a dipole antenna.

The radiation pattern of an antenna is useful in designing Bluetooth products because it tells you what the signal strength will look like from different angles to the product. In the example of a dipole aerial given above, the antenna works best if it is at 90 degrees to the path leading to the device it is communicating with. So if devices are spaced out horizontally, dipoles work best if they are kept vertical. On the other hand, there is still good signal strength at up to a 45 degree angle, so the product could be held at an angle and the aerial would still work. Or, products could be spaced out with say a laptop on a desk and a cellular phone under the desk and there would still be a good chance of the two devices getting a good enough signal to connect.

Some antennas have very directional radiation patterns and others are closer to isotropic (a spherical pattern the same in every direction). The ability of an antenna to concentrate radiation in a particular direction is known as its directive gain.

The desired pattern will vary according to the type of product an antenna is being built to. LAN access points may wish to cover a room and have directional coverage which can be targeted into the room. This is possible for a LAN Access Point, as its antenna can be aimed when it is installed. Conversely, handheld devices such as cellular phones generally need to transmit and receive over a wide range of angles, so you definitely do not want strongly directive gain patterns. When choosing an antenna for a product, the desired radiation pattern suitable for its usage and positioning should be into account.

Finally, the orientation of the electromagnetic waves may be taken into account. This is known as the polarization. Polarization has linear and circular elements. For the best performance, transmit and receive antennas are matched for polarization. This is unlikely to be practical with Bluetooth systems, as such a wide variety of devices will be operating at such a wide variety of angles. A combination of antenna radiation patterns and polarization effects could lead to devices performing better when held at different angles. Ideally the user should not be aware of these effects and Bluetooth should appear to be a completely directionless system.

3.3 Gains and Losses

The gain of an antenna is the ratio of power in to power out. Usually antenna gain is measured in dBi. This is gain relative to an isotropic antenna (an antenna which radiates the same in all directions in a perfect sphere and has a gain of one, so power out=power in).

Severe losses can occur if the feed between the radio and the antenna is not well matched, as the signal will be reflected back down the feeder. Further losses are experienced in the antenna itself and then there are material losses due to the signal being absorbed in the propagation medium.

In the case of microwave transmissions, water is a strong absorber, so signals which have to pass through flesh (mostly water) will suffer high material losses. Losses will also occur through furniture and metals will block the signal.

Figure 3.2 illustrates the losses in getting a signal from the radio transmitter through to the receive antenna.

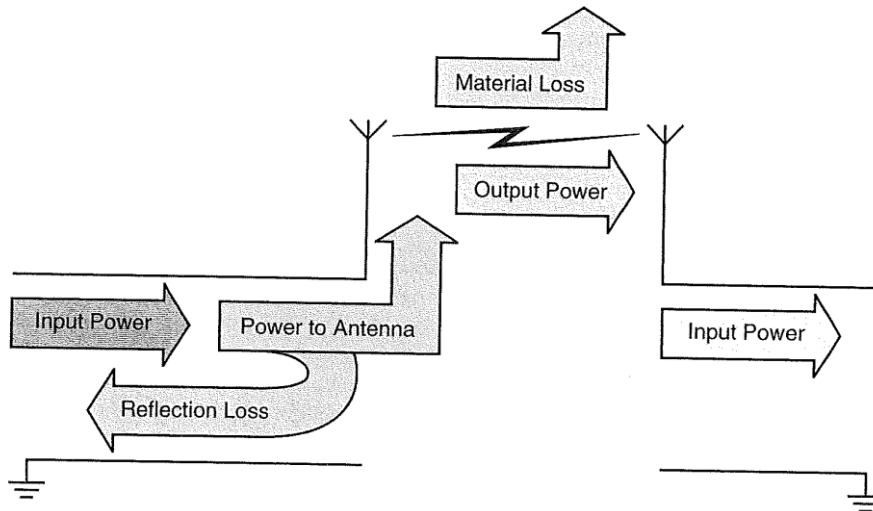


Figure 3.2 : Losses in transmitting through an antenna.

Of course, the material losses in the propagation path will include the casing of the Bluetooth module. Products such as laptop computers are typically manufactured with metal screening around the internal components. The metal screen is designed to contain radio signals, so a Bluetooth antenna could not be placed inside such a screen. This is the reason why Bluetooth PCMCIA cards have bulges protruding outside the laptop. These contain the antenna.

Most plastics do not absorb significantly in the ISM band, so plastic casings should not cause significant material losses for Bluetooth systems.

3.4 Types of Antennas

The most popular antenna types for Bluetooth devices are dipole, flat panel and microstrip. Other antenna types are possible in the ISM band, such as multiple element dipoles, Yagis, parabolic dishes and slotted antennas. However, the more complex antennas are less likely to have uses in Bluetooth systems. This is because of cost, form factor, or because their radiation patterns are strongly directional, which tends not to suit Bluetooth applications.

3.4.1 Dipole Antennas

Dipole antennas are cylinders, with the signal usually feeding in from the bottom. Very simple dipole antennas are often made out of short sections of coaxial cable for development purposes.

As the elevation patterns in Figure 3.1 showed, a dipole antenna transmits best from the side of the antenna. Usually it is fed from the base of the antenna, but it can also be fed from the centre of one side.

The length of a dipole antenna must be related to the wavelength of the signal it is carrying. Half wave and quarter wave dipoles are commonly used.

Dipole antennas are available in various form factors. Figure 3.3 illustrates the variety available with a snap in surface mounted package at the left, and a swivel antenna with SMA connector at the right. Both are half wave dipole antennas.

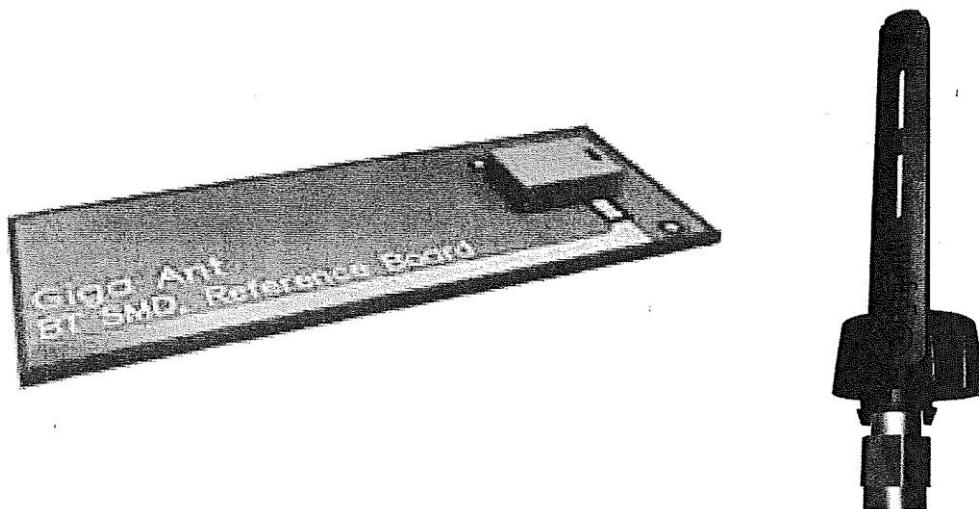


Figure 3.3 : Two types of dipole antenna package.

3.4.2 Flat Panel

Flat panel antennas are small metal patches, and are usually square or rectangular. They are strongly directional, so their radiation pattern is not ideal for handheld devices. On the other hand, they can be made very small for the ISM band and can be mounted directly onto PCBs, both of which help to reduce costs.

A popular form of flat panel antenna used in Bluetooth devices is a PIFA (Planar Inverted F Antenna). This antenna is named for its resemblance to a capital F on its side. It has a flat panel as far away from the ground as possible and is fed by two contacts which form the arms of the F. (see Figure 3.4)

A PIFA antenna can also be fabricated as a microstrip antenna using tracks on a Printed Circuit Board or PCB.

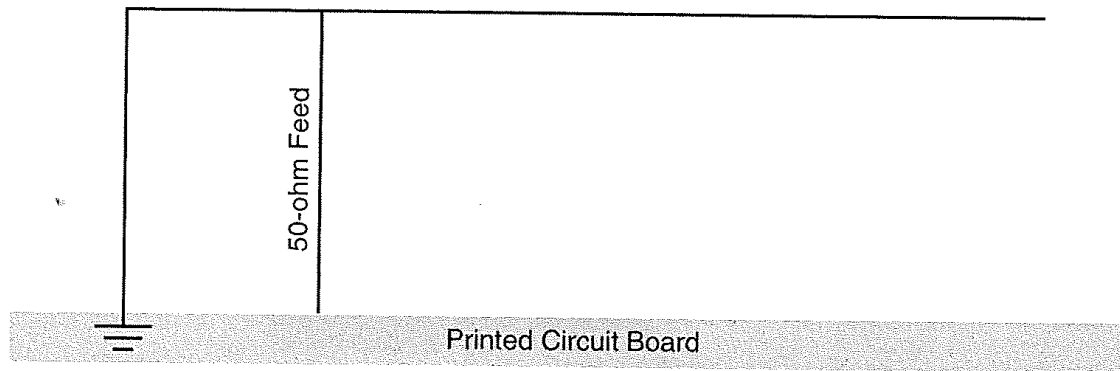


Figure 3.4 : PIFA antenna structure.

3.4.3 Microstrip

Microstrip antennas are simply patterns on PCBs. The fact that tracks on a PCB can be made into a useful antenna illustrates the care that designers must take in product design if they are not to inadvertently produce radiating components where they don't want them.

3.5 Ceramic Antennas

Ceramic antennas can take advantage of physical properties of specialized ceramic materials, such as high permittivity, to produce miniaturized antennas.

Recent developments in ceramic antenna technology are adding functionality and improving performance of antennas. For instance, it is now possible to construct a ceramic antenna which is highly efficient and can be made both directional and steerable (see Figure 3.5). Many antennas are highly affected by their surroundings and perform poorly when placed close to one another, these new antennas experience less proximity effects.

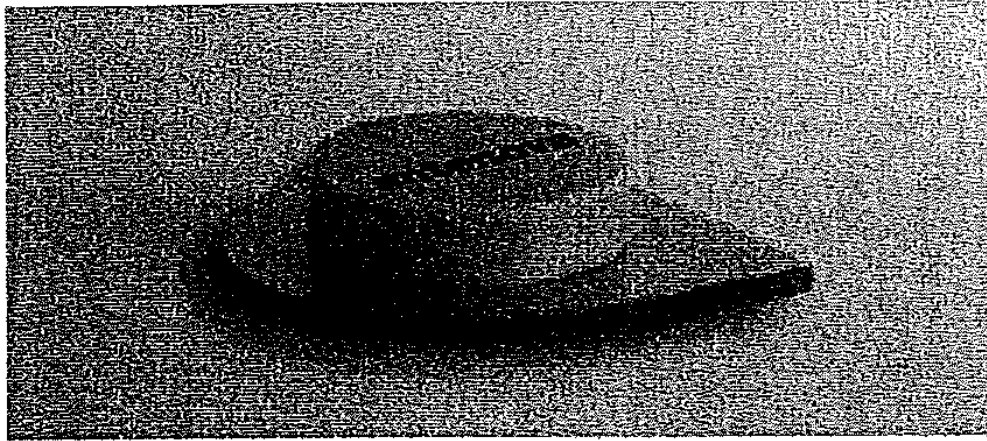


Figure 3.5 : High efficiency ceramic antenna.

3.6 On-chip Antennas

The antenna is implemented as a four armed, spiral microstrip on the top surface of the Bluetooth chip's ceramic package. An RF filter is also printed on the top of the package to improve the characteristics of the device's transceiver (see Figure 3.6).

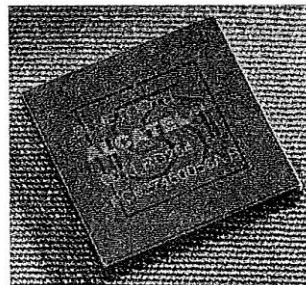


Figure 3.6 : Bluetooth chip with on-chip antenna.

The cost and space savings of such antenna are obvious. However, because the radio signal is blocked by the chip beneath it, the on-chip antenna can only radiate in a hemisphere on one side of the chip.

This might be seen as an advantage for some products such as headsets, where health concerns have been expressed. Users may be able to put up with the inconvenience of only being able to connect to devices on one side of their body for the sake of the assurance that radiation levels will be reduced yet further below the already low levels of Bluetooth. However, there are some products which will require more complete coverage. Therefore, even a device with an on board antenna will typically provide for the alternative of connecting an external antenna.

3.7 Antenna Placement

The characteristics of an antenna are strongly affected by the surrounding ground planes. Shielding from casings and components also affects antenna radiation patterns, and of course, the feed to the antenna affects reflection losses. The combination of these factors means that for optimum radio performance, it is crucial to take the antenna into account when designing PCBs for microwave devices.

4

ENCRYPTION AND SECURITY

4.1 Introduction

Cable based communication is inherently insecure. However, since anyone could potentially listen into a wireless transmission, security is a key issue for wireless communications systems.

Security is dealt with at many levels in the Bluetooth specification:

- The baseband specification details the SAFER+ algorithms used for security procedures.
- The Link Manager specification covers link level procedures for configuring security.
- The HCI specification details how a host controls security and how security-related events are reported by a Bluetooth module to its host.
- The Generic Access Profile covers security modes and user-level procedures for use in all products implementing Bluetooth profiles.
- There is also a Bluetooth SIG white paper on the security architecture, which suggests a framework for implementing security and gives examples of how services might use security.

The Bluetooth specification uses a variant of the SAFER+ cipher to authenticate devices (to ensure they are who they claim to be). Designed by Cylink Corporation as a candidate of the U.S. Advanced Encryption Standard (AES), it has since been released into the public domain.

The encryption engine must be initialized with a random number. After initialization, the encryption engine needs four inputs:

1. A number to be encrypted or decrypted (this is the data being passed between devices.)
2. The master's Bluetooth device address.
3. The master's Bluetooth slot clock (clock bits 26-1; bit 0, which measures half slots is not used.)
4. A secret key which is shared by both devices.

All devices in a piconet know the master's Bluetooth device address and slot clock. The secret key used for encryption varies. Sometimes a device wants to verify that it shares a secret key with another device that claims to share the key. The verifier can't just ask the claimant to transmit the key because anybody could eavesdrop on it. Instead, the verifier sends a random number and gets the claimant to encrypt the number using the secret key and return the encrypted version. The verifier can encrypt the random number using the secret key and compare its result with the claimant's result. If they match, then both sides must have had the same key. This exchange of messages is shown in Figure 4-1.

The full exchange of messages to authenticate a device is slightly more complicated than this, as both devices' encryption engines must first have been initialized with the same random number.

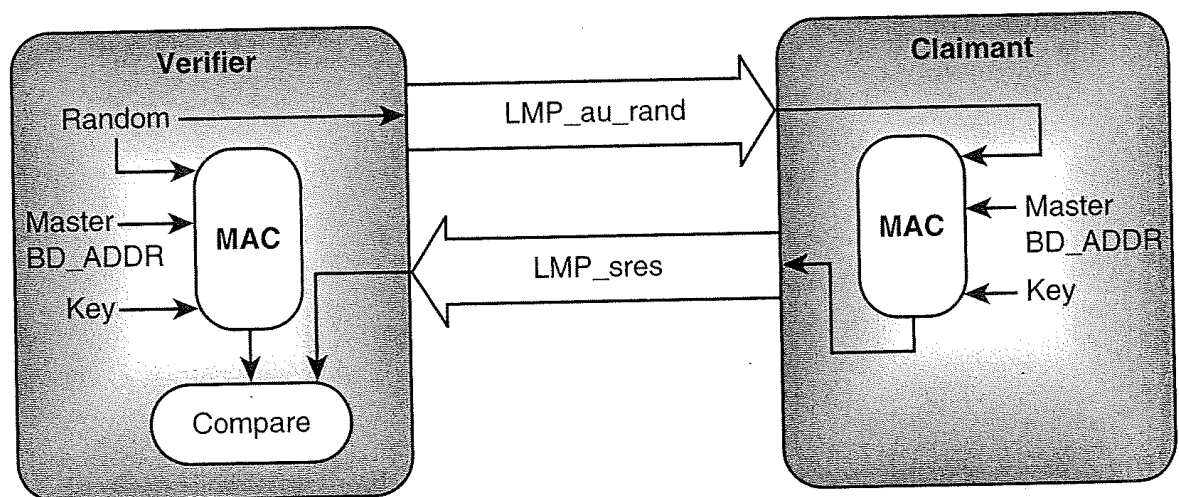


Figure 4.1 : Authentication using the Bluetooth encryption engine.

4.2 Key Generation and the Encryption Engine

The cipher algorithm adopted by the Bluetooth SIG for authentication and encryption is a variant of a strong contemporary algorithm available in the public domain. SAFER+ is the latest in a family of 64 bit block ciphers developed by the Swiss Federal Institute of Technology and Cylink Corporation in the United States since 1993. SAFER+ generates 128 bit cipher keys from a 128 bit plaintext input.

In 1998, SAFER+ was submitted as a candidate successor to the Data Encryption Standard (DES)-referred to as the Advanced Encryption Standard (AES) –in the United States.

During the AES candidate testing phase in 1999, SAFER+ was found by the U.S. National Institute of Standards and Technology (NIST) to have a good security margin with only some minor security gaps. In fact, these do not affect the 129bit version of the algorithm used in Bluetooth anyway. However, it was not accepted

In Bluetooth, the plaintext is provided by a combination of a predefined device PIN number or a unit key and random number. The resulting key is then loaded together with the BD address, Master clock bits, and another 128 bit random number into a bank of Linear Feedback Shift Registers (LFSRs). The output of these LFSRs is combined by a Finite State Machine (FSM) called the “Summation Combiner” to produce a cipher stream which is then exclusive-OR’d with either the transmit or receive data streams as required.

The LFSR block and Summation Combiner are together referred to as the “Encryption Engine” and this process as the “E0” algorithm. This is the part that actually encrypts or decrypts the data bitstream, while the key generator is the part that uses the SAFER+ algorithm to generate the keys used by E0.

The diagram in Figure 4.2 illustrates the functional structure of the authentication and encryption procedures. During initialization, a device specific PIN number is used to generate a 128 bit key using the BD_ADDR of the claimant and a random number shared by the claimant and the verifier. The authentication procedure ensures that both units are using the same 128 bit key, and therefore that the same PIN number was entered into both units. This key K_{init} is used to create a new 128bit key, shared between two units K_{combo} by the key generator which includes the current key, a new random number from each unit and each units’s BD_ADDR. This new key is a link

key and is used with the BD_ADDR and the results of the authenticate routine to produce an encryption key K_c . This encryption key may be shortened to K due to national security export restrictions in some countries. This encryption key is then used with the Bluetooth clock value and the BD_ADDR to initialize the Encryption Engine, which produces the cipher stream. This cipher stream is then used to both cipher and decipher the bitstream data.

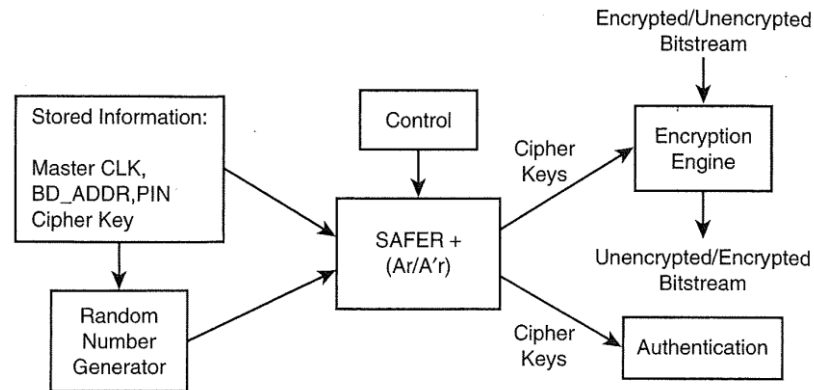


Figure 4.2 : Encryption and authentication block diagram.

There are three main operations that need to be performed:

- Random number generation-This may be carried out in hardware or software.
- Key generation-Based on the SAFER+ algorithm, this is a slow process, typically involving hardware and software elements. Since key generation is not performed frequently and then only during the lengthy LMP negotiations procedures, it is not time critical.
- Encryption-Engine initialization and cipher stream generation use precalculated encryption keys and address/clock information, but the stream generation occurs in real time and is thus typically implemented in hardware.

4.2.1 Encryption Keys

There are a number of different keys used in Bluetooth, and these can be divided into three main types: link keys, sub keys and the resulting encryption keys. Each key is generated using one of a set of five different “E” algorithms and with the exception of E0 they are all based on the SAFER+ algorithm.

- Link Key : K . Link keys are used as authentication keys between Bluetooth devices and to generate encryption keys. There are various types of link keys, each is generated in a different way. Link keys are 128 bit numbers generated using E implementations of the SAFER+ algorithm. They are used for all security transactions. Link keys can be either semipermanent (used for many sessions) or temporary(used during a current session only). Whenever a new link key is generated, it is verified by mutual authentication.
- Master Key: K_{master} . This type of key is for point to multipoint communications and may replace for a time the current link key. This key is generated using an E22 implementation of SAFER+ algorithm and is temporary.
- Unit Key: K_A . This semipermanent key is generated in every single unit often only once during factory setup. While it is unlikely, the unit key might be changed at any time.
- Combination Key: K_{AB} . Changing the unit key is undesirable since in some systems, many units may wish to use the same unit key as link key. A combination key is dependent in two units; each unit produces and sends a random number to the other. A new 128 bit combination key is derived using SAFER+ for each new combination. A combination key replace is often used to replace the unit key for a period and while they are generated in a different way, they are functionally indistinguishable. A combination key is often created toward the end of unit pairing.
- Initialization Key: K_{init} . The 128bit initialization key is a link key used for a single session and is created each time the unit is initialized. The initialization key is only used when combination keys or unit keys have been exchanged yet. The key is generated using an E22 implementation of SAFER+ and uses the PIN number. An initialization key is often created toward the end of unit pairing.

- Encryption Key: K_c . This key is derived from the current link key, but may be shortened due to national security export restrictions in some countries. The full-length key is derived with the E3 SAFER+ algorithm. The Encryption Engine, E0, uses this key to produce the cipher stream.

4.2.2 The E Algorithms

E0: Cipher stream generation/Encryption Engine

E0 creates and applies the cipher stream to the bitstream data.

First the block of LFSRs is loaded with the BD address, Master clock bits and 128 bit random number in an appropriate order. The outputs of these LFSRs are combined by a Finite State Machine called the “Summation Combiner” to produce a cipher stream. This is then exclusive-OR’d with either the transmit or receive data streams as required. The Bluetooth clock, CLK[26:1], is of course incremented on each slot and since E0 is reinitialized at the start of each new packet, a new cipher stream will be created for each packet.

- E1: Authentication. Here, both A_r and A_r' are used to encrypt and validate the E2-generated keys used in the authentication process.
- E2: Authentication key generation. E2 creates the keys which are to be used by the E1 authentication algorithm. Two modes of operation are used depending on the key to be generated:
E21- Uses a 48 bit BD address to create unit keys and combination keys.
E22- Uses a user-supplied PIN to create initialization keys and the master key.
- E3-Encryption key generation. E3 is the algorithm that generates the ciphering key, K_c used by E0. E3 is based on A_r' , the modified SAFER+ algorithm.

All SAFER+ based algorithms-that is E1, E2x and E3- take a 128 bit input and return a 128 bit key. However, to comply with certain national security export restrictions, E0 includes a key length reduction mechanism, which ensures that the LFSRs are loaded with a key of the permissible effective length.

4.2.3 Key Generation and SAFER+

The original SAFER+ algorithm uses a fixed block size of 128 bits, with key lengths of 128, 192 or 256 bits. For Bluetooth, the key length is between 1 and 16 octets, so a Bluetooth key is between 8 and 128 bits. If a key length shorter than 128 bits has been selected, then the key length used for encryption is reduced by a modulo operation. The reduced key is encoded with a block code; this is done to more uniformly distribute the starting states of the encryption sequence.

The SAFER+ algorithm processes the 128 bit input as 16 octets. The algorithm is broken down into 8 rounds, where all 16 octets are processed bit serially into parallel.

For each round, two sub-keys are combined with the new input data. One sub-key is applied to the input data, while the other is applied to the data after the substitution stage. In both cases, the sub-key elements are added both bitwise and octetwise. After the last round, a seventeenth sub-key is also applied, this time to the result data. Each of the sub-keys is created from the input word according to a schedule, which is dictated by the “Bias Words”. This serves to randomize the sub-keys produced.

Each round consists of two “substitution” functions: one that implements an exponential function and one that implements a logarithmic function. These introduce the desired nonlinearity.

An Invertible Linear Transform is then imposed in the form of a Pseudo Hadamard Transformation, followed by an Armenian Shuffle bitwise interleaving function. These two operations are carried out three times with a final PHT phase at the end. The PHT function consists of multiple accumulates and bit shifting operations.

The seventeen sub-keys are generated from the 128 bit input to the algorithm. The Sub-Key Generation process involves creating a parity word, rotating each of the octet bits and rotating the octets. The result is then added mod256 to a precalculated bias word.

The block diagram in Figure 4.3 depicts the basic structure of the algorithm. Look-up tables are shown for the log, exponent and bias functions, which is the most likely implementation, though the actual function could of course be used if appropriate.

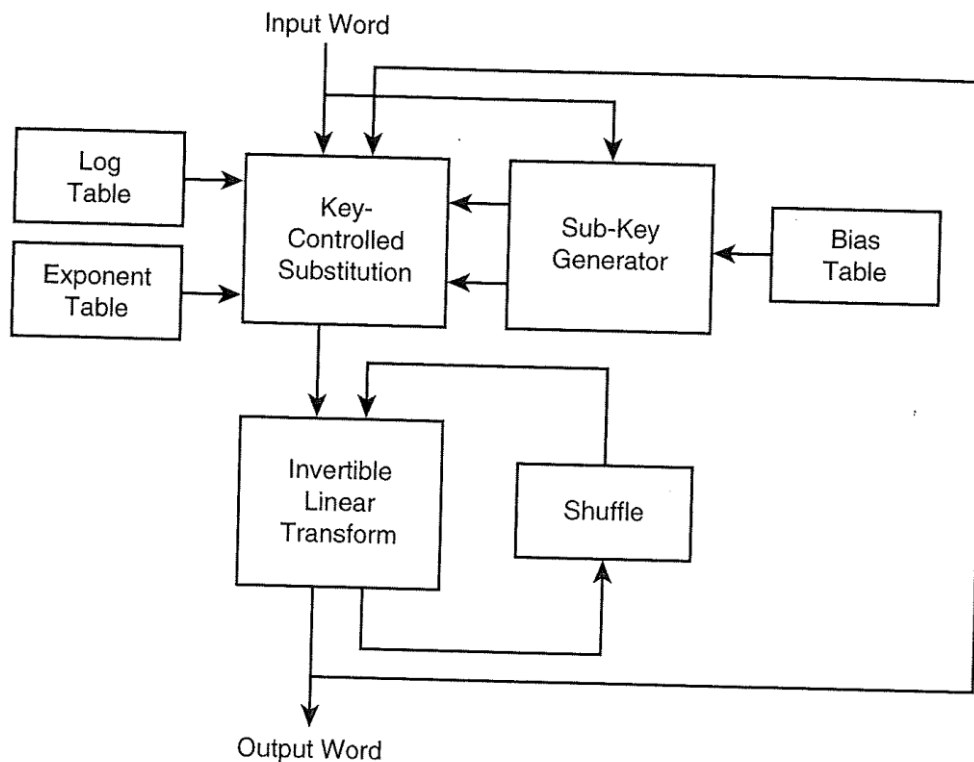


Figure 4.3 : SAFER+ functional block diagram.

4.2.3.1 Ar and A'r

The SAFER+ algorithm is referred to as Ar in the Bluetooth standard. However, as such, it is only used as part of the authentication procedure. The A'r algorithm is used at least once in almost all key generation procedures and is a modified version of the SAFER+ algorithm where the input to Round 1 is fed back into the algorithm during Round 3. This makes A'r noninvertible and of course unsuitable for use as an encryption algorithm.

4.2.3.2 Advantages of SAFER+ and Associated Implementation Issues

In cryptographic terms, SAFER+ is a relatively simple algorithm, yet it provides a high level of security. Its designers claim that it has no weak keys, is robust against both linear and differential cryptanalysis, and its transparency(that is, its use of only well defined mathematical functions) makes it clear that there are no so called “trap doors” allowing third party deciphering. The minor security gaps mentioned above were uncovered using differential cryptanalysis and affect the 192 and 256 bit versions of SAFER+. The 128 bit version as used in Bluetooth diffuses the full key

into the algorithm very quickly and so is robust to such attacks. In addition, regular link key changes will further prevent the viability of a cryptanalysis attack.

Its regular structure and byte orientation make the algorithm suitable for implementation in silicon and small footprint microprocessors, while also being highly optimisable for modern high performance DSPs or 32 bit microprocessors. A silicon implementation of the SAFER+ algorithm can compute a 128 bit key in less than 100 μ s when clocked at 20 MHz.

The following sections explain how the SAFER+ encryption engine can be used to support a variety of security features.

4.3 Secret Keys and Pins

To use encryption, Master and Slave must share the same secret key. This secret key is never transmitted on air. The secret key could be built in by manufacturers (a fixed key), or it could be derived from a Personal Identification Number(PIN) entered through a user interface(a variable key).

An example of a device which could sensibly use fixed keys is a headset for a cellular phone. These could be sold with fixed keys, so that they would not need a costly and bulky user interface to enter security information. To ensure that both ends of the link share the same keys, the user could enter the headset's information into a cellular handset(these already have an interface suitable for entering numbers, so, unlike the headset, a facility to enter PINS would not add to the cost of the device).

An example of an application where PINS might need to be altered frequently is a hotel or conference center offering Bluetooth access points. When a guest checked in, he or she could be given a PIN number which would allow use of encryption on data sent to the LAN access points.

If a device is to have variable PINS, then naturally the user interface must support entering new PINS. So for devices with an HCI, it is the host (which owns the user interface) that determines whether the PIN is fixed or variable. The HCI_Write_PIN_Type command is used by the host to tell the Bluetooth device whether the PIN is fixed or variable. (The HCI_Write_PIN_Type command can be used to check whether the lower layers believe a fixed or variable PIN is in use.)

When a Bluetooth device needs to query the host for a PIN, it can send the event HCI_PIN_Code_Request_Event. If the host can supply a PIN, it replies with the

command `HCI_PIN_Code_Request_Reply`, which contains the PIN in its parameter list. If the host has no PIN to supply, it responds with the command `HCI_PIN_Code_Negative_Request_Reply`, which will cause attempts at using security features to fail.

4.3.1 The Bluetooth Passkey

The Generic Access Profile defines the terms used by a Bluetooth device's user interface. HCI and LMP use the term "PIN", but the Generic Access Profile requires the user interface to use the term "Bluetooth passkey".

The PIN used by the baseband can be up to 128 bits. PINs can be entered as decimal digits, or optionally they may be entered as alphanumeric characters. Unicode UTF-8 coding is used to transform the characters into digits.

Because some devices which allow PINs to be entered will not support alphanumeric entry, devices sold with fixed PINs should be sold with a note of the PIN given as decimal digits.

The Logical Link Control and Adaptation Layer (L2CA) needs to be aware that entering PINs through a user interface may take some time. L2CA has a timeout on a response (RTX). The RTX timer's value is implementation dependent, but it is initially set between 1 and 60 seconds. If the timer elapses while waiting for PIN entry, the times out request will be resent with the timeout doubled. This continues until the requester decides to abandon configuration. To avoid this timing out, a device which knows it will take some time should send a connection pending response to its peer. This indicates that some processing is happening which may take some time and causes an Extended Response Timer(ERTX) to be starts in place of the RTX timer, thus giving sufficient time for the PIN to be entered. ERTX is again implementation dependent, but its value is initially between 1 minute and 5 minutes, so it allows much more time for the user to enter PIN.

4.4 Pairing and Bonding

The Generic Access Profile calls two devices that know they share a link key bonded. The procedure involved in creating a relationship based on a common key link is called bonding.

Bonding involves creating a link specifically for the purpose of creating and exchanging a common link key. During bonding, the link managers create and exchange a link key then verify it by mutual authentication. The Link Level procedures of link key generation and authentication as shown in Figure 4.4 are collectively called pairing.

Bonding may involve higher layer initialization procedures as well as link level pairing. At the User Interface Level, the term Bluetooth bonding is used to refer collectively to bonding and pairing procedures.

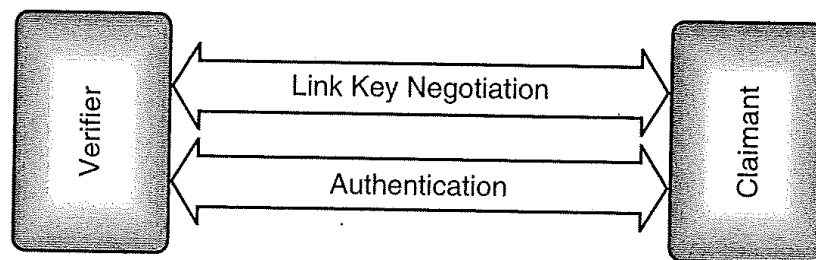


Figure 4.4 : LMP procedures involved in pairing.

4.4.1 Authentication

Authentication is the process by which devices verify that they share the same link key.

Mutual authentication takes place when link keys are generated; authentication can also be controlled using HCI commands. The process of authentication itself uses a series of messages to be exchanged using Link Manager Protocol.

Authentication can be triggered via HCI commands at any time; it does not have to happen at link setup. For instance, a new application which requires security might start using an existing link. This would trigger authentication.

Authentication would usually take place as a prelude to setting up encryption on a link, but authentication can be done independently of encryption. It is conceivable that a device might want to use authentication to check if it is communicating with the correct device, even if it had chosen not to encrypt traffic on the link.

For devices with an HCI, authentication can be requested with the command `HCI_Authentication_Requested`. When authentication completes, the `HCI_Authentication_Complete` event is sent from the device to the host. This contains

a status field, which either indicates success or failure. Possible reasons for failure are:

- The connection being authenticated doesn't exist.
- Authentication failed.
- Authentication isn't a supported feature on the Bluetooth device.
- The command is not allowed (for example, when authentication has been disabled).

Authentication can also fail if the claimant does not have a link key to authenticate with. In this case the claimant responds to the LMP_au_rand with LMP_not_accepted.

As a side effect of a successful authentication procedure, a parameter called Authenticated Ciphering Offset (ACO) is calculated and then used to generate the ciphering keys, which are then used to encrypt data. If a master and a slave initiated authentication together, they could end up with two different ACOs, so they would not be capable of decrypting one another's encrypted data. To keep this from happening, version 1.1 introduced a new rule that link managers must reply to any outstanding LMP_au_rand authentication request signals with LMP_sres secure response signals before sending their own LMP_au_rand authentication request signals. It is still possible for LMP_au_rand messages to cross, however. If this happens and the Master receives a response to its own LMP_au_rand, it is allowed to respond with LMP_not_accepted with the error code "LMP Error Transaction Collision". In this way, Link Managers should be capable of ensuring that only one authentication is in progress at any time, thus also avoiding mismatching ACO's.

Authentication can be enabled or disabled via the HCI using the command HCI_Write_Authentication_Enable. HCI_Write_Authentication_Enable can be used to check whether authentication is enabled or disabled. Authentication cannot be enabled on a per-connection basis; It is either enabled or disabled on all connections at once.

Before authentication can take place, both devices must initialize their encryption engines with the same number, this process is shown in Figure 4.5. An LMP_in_rand

message is sent carrying the random number; both sides then use the number to initialize their encryption engines.

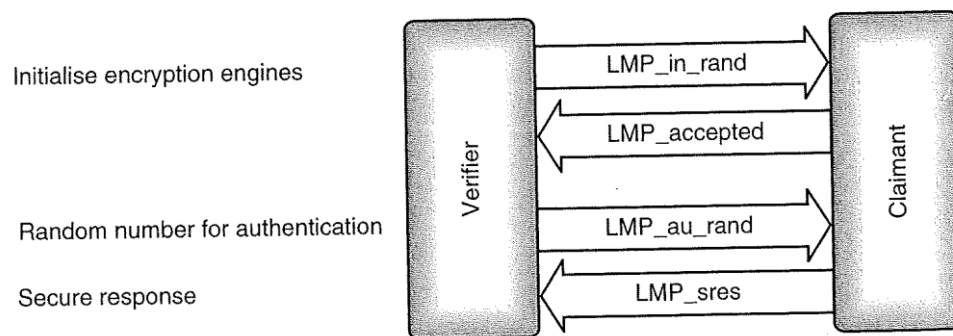


Figure 4.5 : LMP message sequence chart for authentication.

Next the verifier sends an LMP_aud_rand message containing the random number to be authenticated by the claimant. The claimant encrypts this number using its link key, and then returns the encrypted number in a secure response message, LMP_sres. The verifier encrypts the random number from LMP-aud_rand with its link key and compares it with the encrypted version in LMP_sres. Thus the verifier can decide whether both sides share the same link key without the link key ever being transmitted on air.

4.4.2 Unit Keys

Every Bluetooth device that supports security has a unit key. The unit creates the unit key using its random number generator on first startup; thereafter, the unit key normally does not change. The unit key is used when generating link keys for secure communications.

If a Bluetooth is sold or otherwise changes hands, the new owner might want to change the unit key. For devices with an HCI, this is simply done by sending the HCI_Create_New_Unit_Key command. If the old key is in use, the old key carries on being used for existing links. So for maximum security, old link keys should be deleted when a new key is created. The host does not need to know the unit key, so there are no messages for a host to read or write the unit key.

4.4.3 Link Key Generation

Once master and slave know that they share a secret key, they could use that key for encrypting traffic. But if data with a pattern is sent, then it is possible to eventually crack the link key. Therefore for maximum security, the link key should be changed regularly. So a mechanism is needed to create link keys to use for data encryption. Obviously a key that was just transmitted on the air would not be very secure, so keys are disguised by exclusive ORing them with a key generated from the random number in the LMP_au_rand message previously sent and the PIN.

To get a shared key, each unit sends a key in an LMP_unit_key or LMP_comb_key message as shown in Figure 4.6. The rules for choosing a key are:

- If both devices send a LMP_unit_key, the Master's unit key is used.
- If one device sends a LMP_unit_key and one sends a LMP_comb_key, the unit key is used.
- If both devices send a LMP_comb_key, then a combination key formed from two keys is used.

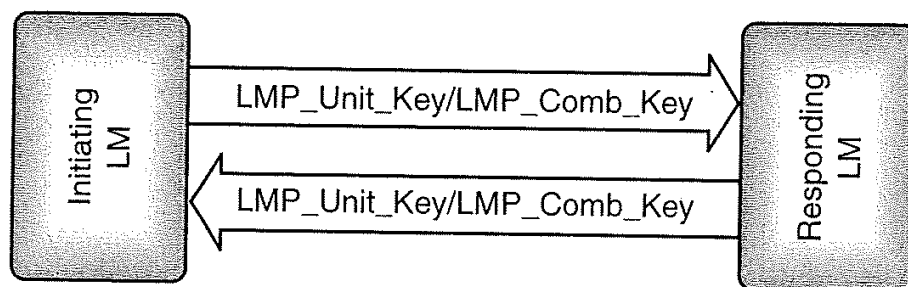


Figure 4.6 : LMP message sequence chart for link key generation.

So a link key can be a unit key chosen by one unit only, or a combination key made of elements from both units. Since it is possible that either device's unit keys may have been compromised, the combination key is more secure and is recommended.

After generation of the link key, both devices mutually authenticate one another by exchanging LMP_au_rand and LMP_sres messages. First the initiating LM sends LMP_au_rand and the responding LM sends LMP_sres, then the responding LM sends LMP_au_rand and the initiating LM sends LMP_sres.

An example of a set of LMP messages used to initialize encryption, generate combination keys and authenticate is shown in Figure 4.7. It is worth noting that the messages exchanged during this process changed between version 1.0b and version 1.1: in version 1.0b authentication took place after the generation of initialization keys, as well as after generation of link keys. Authentication twice slowed down the process and did not increase security, so the duplicate authentication was removed from version 1.1 of the Bluetooth specification.

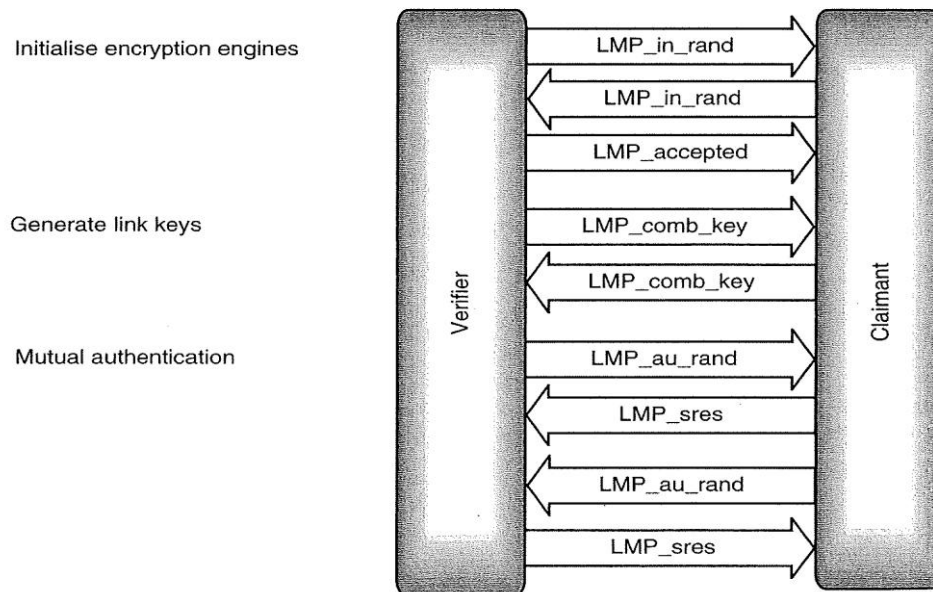


Figure 4.7 : LMP message sequence chart for authentication.

4.4.4 Changing Link Keys

If the host decides for some reason that the current link key may have been compromised, it can create a new link key using the HCI command `HCI_Change_Connection_Link_Key`. Because each connection uses a different link key, this command has a `connectionHandle` parameter to identify the connection on which the link key is to be changed. It may take some time for a new link key to be negotiated, so the Bluetooth module replies to this command with an `HCI_Command_Status` event.

The sequence of LPM messages used to change the link key is shown in Figure 4.8. It is exactly the same as the messages used to negotiate the key in the first place. If the key is a unit key, it can not be changed at the LMP level, so when a new combination key is sent, it will be rejected with an `LMP_not_accepted` message.

Once the new link key has been generated, an HCI-Link_Key_Notification event and HCI_Change_Connection_Link_Key_Complete event are sent to the host. Both devices also conduct mutual authentication after changing the link keys.

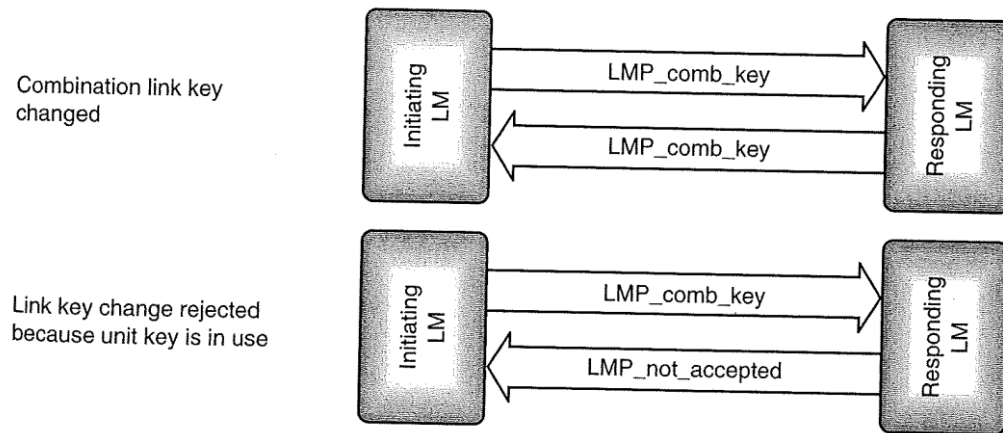


Figure 4.8 : Message sequence charts for changing a link key.

4.4.5 Changing to Temporary Link Keys

If broadcast information is to be encrypted, a temporary link key must be used. A temporary key is needed because when a device receives a packet, it must decrypt it immediately so that it can respond to any errors in the packet. A Slave device does not know until it receives a packet whether it is broadcast or point to point and so does not have time to switch between a broadcast and a link key. Since there is no time to switch keys, the device must use the same key for broadcast and point to point links.

Because broadcasts are sent to every device in the piconet, the same broadcast link key must be used by all devices on the piconet. This means that devices which previously had individual link keys and could not read one another's packets are now all using the same key, so security is compromised. Furthermore, the link key must be usable by all devices, so it must use the shortest key length of any of the devices in the piconet. Obviously if security has to be compromised in this way to implement broadcast encryption, the links should return to using their normal link keys as soon as broadcast encryption is switched off again. Therefore, the link key used for broadcast is a temporary key.

Because only the Master can broadcast, it is the Master that creates the temporary link key. An HCI_Master_Link_Key command can be used by the host to force a Master to create and use a temporary link key. This command has a Key_Flag

parameter which is used to specify the type of link key being created. Because some link management level negotiation must take place before the keys are in use, the module responds immediately with an HCI_Command_status event and only sends an HCI_Master_Link_Key_Complete event when all LMP negotiations have taken place and the new key is in use. Both devices conduct mutual authentication to verify the new link key.

The temporary link key will only be valid for the current session, so every time a new encrypted session is started, a new temporary link key will need to be created (and mutual authentication is conducted to verify the key). Because the temporary link key is only used for a short period, temporary link keys cannot be changed (in version 1.0b any keys could be changed, but in practice temporary link keys never needed to be changed, version 1.1 removed the ability to change temporary link keys).

To create a temporary link key, the Master first creates a 128 bit master key, K_{master}. The Master creates this key by combining two 128 bit random numbers using the SAFER+ Encryption Engine. This Encryption Engine is used instead of using a random number directly in case the Master's random number generator is not very good. By combining two random numbers in this way, an extra degree of randomness is introduced, making it much more difficult for a snooping device to guess the master key.

Having created the master key, the Master creates another random number and sends it to the Slave in an LMP_temo_rand_message. Both Master and Slave then use the SAFER+ Encryption Engine to combine the random number and the current link key to create an overlay. The Master adds this overlay modulo-2 to the master key and sends the result in LMP_temp_key as shown in Figure 4.9.

Because the Slave calculates the same overlay, it can extract the master key. Thus, as soon as it receives LMP_temp_key, it extracts the master key, mutual authentication takes place to verify the key, and it is then used as the current link key. Every time the link key is changed, encryption is stopped and restarted to ensure that all devices on the piconet have picked up the new key and are using the correct parameters.

The master key carries on being used until the end of the session, or until the link key is changed again.

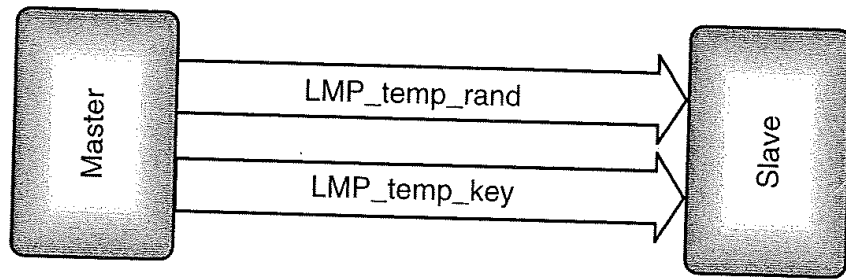


Figure 4.9 : Message sequence chart for changing to a temporary link key.

4.4.6 Reverting to Semipermanent Link Keys

The semipermanent link keys are just the normal link keys used for point to point communications. For some devices, such as headsets, there may be no facility to enter new keys. For these devices the term “semipermanent” may be misleading, as the key used for point to point communications is permanently stored. For other devices the key may occasionally be changed and the term “semipermanent” is more accurate. The same HCI_Master_Link_Key command that was used to switch to a temporary key is used to switch back to a semipermanent key. Only the Key_Flag parameter is changed to specify that the key is reverting back to the semipermanent link key which was in use before the temporary link key.

Because the semipermanent link key is the link key which was in use before, both devices already know the key. This means that there is no need to send the key, so the LMP_use_semi_permanent_key message has no parameters. The Slave cannot refuse a request to return to using the semipermanent link key, so it simply acknowledges receipt of the message with LMP_accepted as shown in Figure 4.10.

As for all other link key changes, when the piconet reverts to using the semipermanent link key, encryption must be stopped and restarted. The device which sent LMP_use_semi_permanent_key initiates authentication once encryption is back on to verify the new link key (arguably this check is redundant as it could tell the key was correct by successful decryption of data).

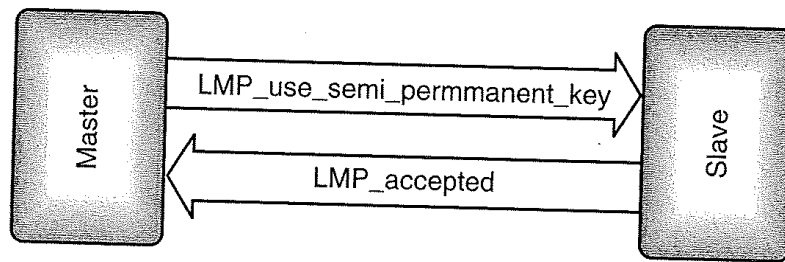


Figure 4.10 : Message sequence chart for changing to a semipermanent link key.

4.4.7 Storing Link Keys

In the procedure described above, a link key was created by negotiation. Link keys can also be set up by simply writing via the HCI, or the keys from one session can be read by the host, stored and then written back later. Remembering link keys from previous sessions can obviously save the time involved in negotiation and get an encrypted link running faster. Many hosts have nonvolatile memory available, so having the host store data between sessions saves adding cost to the Bluetooth device. Another advantage of the host remembering keys is that if the host is something like a laptop with Bluetooth PCM-CIA card, changing cards will not cause keys to change. Also, if the card is removed, security keys cannot be read from it. Keys stored on the host can be protected by passwords and so are potentially more secure than keys stored in a removable Bluetooth device.

Every time a new link key is generated, an HCI_Link_Key_Notification event is sent to the host. The parameters of this message are the new link key and the Bluetooth Device Address (BD_ADDR) of the device at the other end of the connection.

When the module wants to retrieve a link key from the host, it sends an HCI_Link_Key_Request event. This event has a single parameter: the Bluetooth Device Address of the device at the other end of the ACL link for which the link key is required. If the host can supply the link key, it is sent back in an HCI_Link_Key_Request_Reply command; if for some reason the host can't supply a link key, it responds instead with a HCI_Link_Key_Request_Negative_Reply command.

The Bluetooth module does not remember link keys when power cycled. Since it tells the host every time a new link key is generated, the host should know all the link keys in use, but it is possible that a Bluetooth module which has its own power source

may be connected to a new host, then the host would not know the keys in the module.

The host is provided with an `HCI_Read_Stored_Link_Key` command to retrieve link keys from the module. The module responds with the `HCI_Return_Link_Keys` event; this event has three parameters:

- `Num_Keys`- The number of link keys being requested.
- `BD_ADDR[i]`- An array of `NUM_Keys` Bluetooth Device Addresses.
- `Link_Keys[i]`- An array of link keys which match the Bluetooth Device Addresses.

The command `HCI_Read_Stored_Link_Key` can be used to read the key for a particular link, or to read all link keys. `HCI_Write_Stored_Link_Key` is used to store a key for a given link (the link is specified by the Bluetooth Device Address of the device at the other end of the link). A device may only be able to store a limited number of keys, so `HCI_Delete_Stored_Link_Key` can be used to remove link keys from storage.

4.4.8 General and Dedicated Bonding

Bonding involves setting up a link for the purpose of exchanging link keys, and possibly other security information. Because the device which initiates bonding is the device which sets up the connection by paging, when bonding, it is always the paging device which initiates authentication procedures.

The Generic Access Profile divides bonding into two procedures: general bonding and dedicated bonding. Dedicated bonding happens when devices only create and exchange a link key. As soon as Link Level authentication procedures have completed, the channel is released before the higher layers connect. General bonding may involve exchange of data by higher layers to initialize their security parameters.

Link keys for bonded devices are stored by a Bluetooth device so that it does not have to create new link keys every time it connects. Since bonding involves creating a new link key, any old link key for the device being bonded with is deleted before bonding is performed. On devices with an HCI, the host can force deletion of a link key using the `HCI_Delete_Stored_Link_Key` command.

Because bonding involves pairing, the paged device must be in pairable mode before bonding can take place.

Figure 4.11 shows dedicated bonding. Note that there is no connection made above link manager level.

As Figure 4.12 shows, general bonding involves all the same steps as dedicated bonding, but in addition an L2CAP channel is set up and depending on the application requiring security, higher layer channels may also be set up. Once such channels are set up, security information from higher layers may be passed across the channel. After higher layers are configured, the connection is torn down again.

Once two devices are bonded, they share a link key and can connect using that link key without having to go through pairing procedures again.

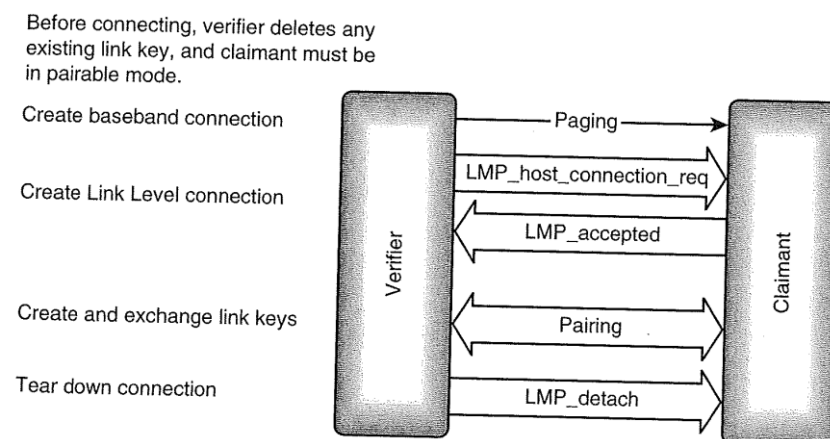


Figure 4.11 : Dedicated bonding.

Before connecting, verifier deletes any existing link key, and claimant must be in pairable mode.

Create baseband connection

Create Link Level connection

Create and exchange link keys

Configure higher layers

Tear down connection

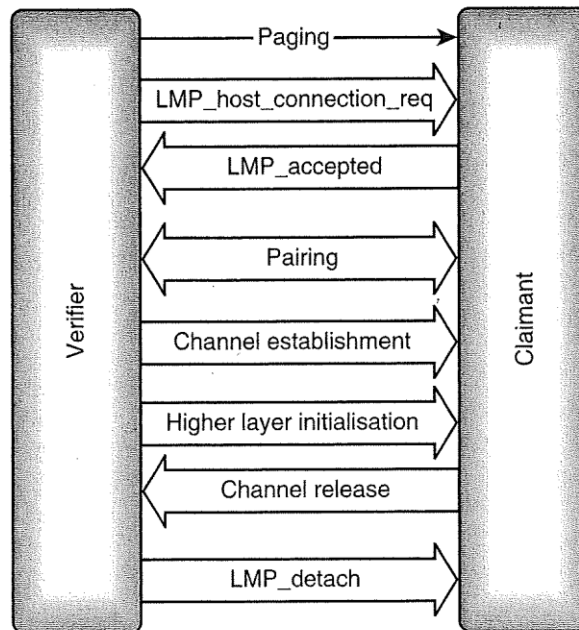


Figure 4.12 : General bonding.

4.5 Starting Encryption

Once two Bluetooth devices have undergone authentication and agreed on a link key, there are three more steps before encrypted traffic can be exchanged:

- Negotiating encryption mode.
- Negotiating key size.
- Starting encryption.

The messages exchanged to start encryption are shown in figure 15.13.

4.5.1 Negotiating Encryption Mode

The encryption mode can be anyone of the following:

- No encryption.
- Encrypt both point to point and broadcast packets.
- Only encrypt point to point packets.

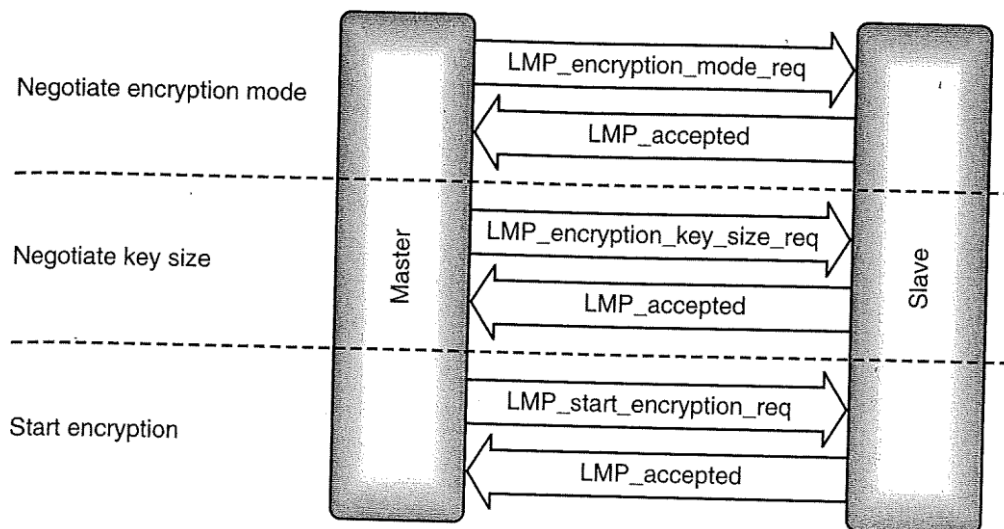


Figure 4.13 : LMP message sequence chart for starting encryption.

On devices with an HCI, the encryption mode can be set using an HCI_Write_Encryption_Mode command and can be checked at any time using the HCI_Read_Encryption_Mode command.

The Link Manager uses an LMP_encryption_mode_req to request that the desired encryption mode be used on the channel. If the encryption mode is accepted, LMP_accepted is sent back. If it is not, LMP_not_accepted is sent, and the Master is free to try again, requesting a different encryption mode.

The encryption mode can be changed by resending the LMP_encryption_mode_req. Data transmission must be stopped when encryption mode changes. Otherwise, data could be sent when the encryption mode is indeterminate, which would lead to data being corrupted or lost. To avoid this, data transmission is stopped before any encryption change. Version 1.0b of the specification said that data traffic had to be temporarily stopped, but there was some disagreement among implementers as to when to stop it: Should it stop immediately, at the end of an ACL packet, or at the end of an L2CAP packet? Version 1.1 of the specification clarified that data transmission should stop at the end of the current ACL packet with L2CAP data.

4.5.2 Negotiating Key Size

The United States has regulations governing the export of devices capable of using strong encryption schemes for encrypting data. To comply with these regulations, it is possible to manufacture a Bluetooth device which will not use the full 128 bit keys for

encrypting data. Before encryption can be switched on, both units must agree on a key length to use. The Master begins by requesting the maximum key length it can use. If it is within the capabilities of the Slave, an LMP_accepted is returned. Otherwise, an LMP_not_accepted is returned and the Master must try again with a shorter key. The Master keeps trying until it gets an LMP_accepted.

4.5.3 Starting Encryption

Once an encryption mode and key size has been chosen for the link, encryption can be switched on and off. The HCI_Set_Connection_Encryption command uses an ACL connection handle to identify which is having encryption switched on or off. When the link manager has finished negotiating encryption on the link, an HCI_Encryption_Change event is sent back to the host. No traffic should be sent on the ACL link while encryption is being enabled or disabled, as the link will be occupied with LMP traffic.

The final step is to send an LMP_start_encryption_req. Once the LMP_accepted reply has been received, encrypted data can be exchanged on the ACL link.

This section has described the Master driving encryption mode, but it is also possible for the slave to send the messages to authenticate pair, negotiate modes, and switch encryption on and off. We have also described a sequence where each message is exchanged in sequence at link setup, but it is equally possible for authentication to proceed at any time, and for link keys to be changed at any time, or indeed link keys from a previous encryption session could be stored and used.

4.5.4 Stopping Encryption

Since encryption does not slow down traffic on a link usually once encryption has been started there is no need to stop it. However there may be a need to change encryption parameters after encryption has been started and this cannot be done while encryption is active. Before making changes which affect encryption-for example, changing link keys- encryption is stopped and it is then restarted after the change.

In version 1.0b, the master could stop encryption with an LMP_stop_encryption_mode_req, but there was not clearly defines way for the slave to stop encryption. Version 1.1 specified that any unit wanting to stop encryption could send an LMP_encryption_mode_req with the encryption-mode parameter set to

no encryption (zero). If the other device responds with LMP_accepted, the Master sends an LMP_stop_encryption_req message to stop encryption.

After any changes are made, encryption can be restarted using the same LMP_encryption_mode_req which was used to initially start encryption.

4.6 Security Modes

The Generic Access Profile defines three security modes:

- Security mode 1 is nonsecure—Devices in Security Mode 1 will never initiate any security procedure. Supporting authentication is optional for devices which only support Security Mode 1.
- Security Mode 2 gives Service Level-enforced security—The channel or service using an L2CAP connection decides whether or not security is required. So until an L2CAP channel has been established, a device in security Mode 2 will not initiate any security procedures. Once an L2CAP channel has been established, the device then decides whether or not it needs authorization, authentication, and encryption, and goes through appropriate security procedures.
- Security Mode 3 is Link Level-enforced security—A device in Security Mode 3 initiates security procedures before it sends an LMP_setup_complete message. If security measures fail the device, the connection will not be setup. It is possible to set up devices supporting Security Mode 3 so that they will only connect with pre-paired devices. In this case, they would reject an LMP_host_connection_req from any other devices (they would reply with an LMP_not_accepted message).

In addition to these specific security modes, the other modes of a Bluetooth device may be used to increase security. For maximum protection of data, a device can be set

in nonconnectable mode when it is not in use. In this mode, the device will not respond to paging, so other devices cannot connect with it.

Nondiscoverable mode can be used to stop a device from responding to inquiries. If this is used, then only devices which already know the device's Bluetooth Device Address can connect to it.

4.7 Security Architecture

The Bluetooth security white paper defines a security architecture which may be used to implement Mode 2 Service Level-enforced security on Bluetooth devices. Because the implementation of security at Service Level does not affect interoperability, the white paper is purely advisory and is not a Bluetooth specification.

4.7.1 Security Levels

In addition to the authentication procedures defined in the Bluetooth specification, the security white paper introduces the concept of an authorized or trusted device. An authorized device has been specifically marked in a server's database as having access to a service.

Devices and services can be divided into different security levels. The security white paper splits devices into three categories and two trust levels:

- Trusted devices- Paired or bonded devices which are marked in a database as trusted and can be given unrestricted access to all services.
- Known untrusted devices- Devices which have been paired or bonded but are not marked in a database as trusted; access to services may be restricted.
- Unknown devices- No security information is stored, the device is untrusted and access to services may be restricted.

It would also be possible to implement different levels of trust for services as well as devices. For example, reading and writing to a calendar could be defines as different services. Read access to the calendar might be restricted to a range of devices known to belong to co-workers who had an interest to seeing appointments.

Write access to the calendar might be restricted to a smaller set of devices belonging to the owner of the calendar.

The security white paper suggests that the security requirements authorization, authentication and encryption of services could be set separately. This gives three security levels for services:

- Open services- Any device may access these; there are no security requirements.
- Authentication-only services- Any device which can go through authentication may access there. (authentication proves it shares a secret key with the service provider).
- Authentication and authorization services- Only trusted devices may access these (trusted devices are recorded as trusted in the server's database as well as having a secret key).

Each service should have its security level set independently, so a device having access to one service does not imply that it has access to others. It should be possible to define a default level of security which will apply to all services, unless they are specifically set to a different level.

4.7.2 The Security Manager

The existence of trusted devices and of different levels of authorization for different services imposes a requirement for databases to hold device and service information.

Different protocols will wish to access the information in these databases according to the profile being implemented; For instance:

- L2CAP will enforce security for cordless telephony.
- RFCOMM will enforce security for dialup networking.
- OBEX will use its own security policy for file transfer and synchronization.

To allow uniform access to the databases by all layers, a security manager handles security transactions with the various layers. All exchange of information with the security databases goes through the security manager as illustrated in Figure 4.14.

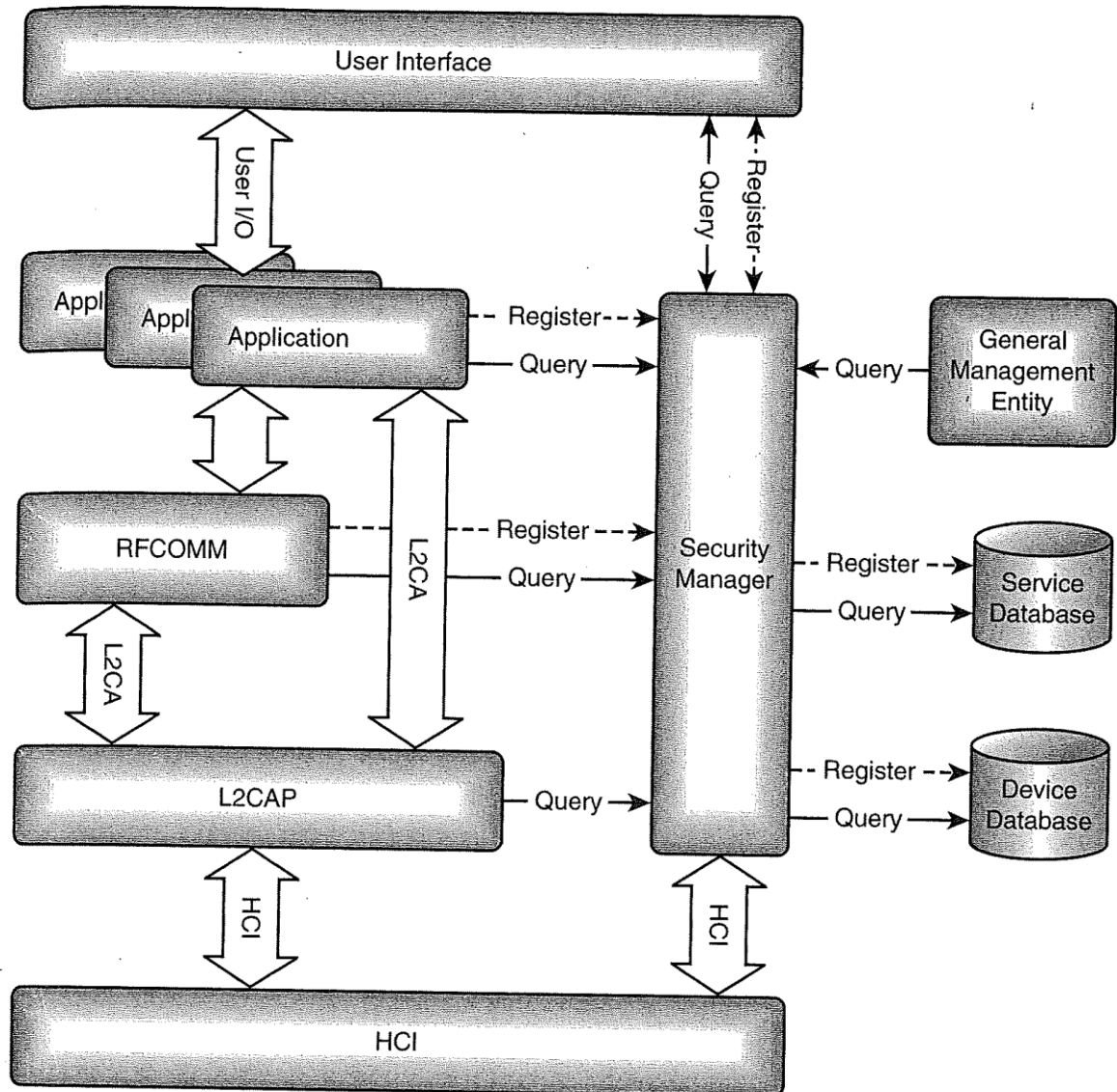


Figure 4.14 : Security architecture.

Applications and protocols wishing to use security features register with the security manager. The security manager stores security information in the security databases on behalf of the rest of the system. Security policies are enforced by exchanging queries with the security manager:

- Applications query to find out whether a particular device is allowed to access a service.

- HCI queries to find out whether to apply authentication and/or encryption to a connection.
- The user interface is queried by the security manager to get PINs.
- The user interface is queried by the security manager to authorize new devices.
- Protocol layers query the security manager with access requests.

The device database holds information on whether devices are authenticated and authorized. The service database holds information on whether authorization, authentication and encryption are required for access to a service.

The security white paper suggests that if a service has not registered with the service database, then the default settings should be:

- Incoming connection-Requires authorization and authentication.
- Outgoing connection- Requires authentication.

As the white security paper is not a part of Bluetooth specification, there is no requirement to implement security in the way suggested by the white paper. However, designers implementing security in Bluetooth devices should consider the white paper's recommendations.

4.7.3 Setting up Security on New Connections

Because it is the service that decides the level of security to be enforced, security cannot be enforced when an ACL (data) connection is first set up. Instead, security is enforced only when access is requested to a protocol or service which requires security. The protocol or service requests access from the security manager. The security manager looks up the service or protocol in the service database to see what level of security to impose. Then it looks up the connecting device in the device database to see whether it meets the requirements of the service. If necessary, the security manager enforces authentication and/or encryption and sends any necessary queries for PINs or authorization to the user interface. Access is then granted or refused and if access was granted the service can be used.

It is possible that some services may use a connection without encryption, then another service will begin using the service which requires encryption. Encryption will be set up for the service which requires it, but other than a short pause in traffic while LMP messages are exchanged, this will not be apparent to the other services.

Other than the link management messages required to configure security, there is no impact on bandwidth. The same number of bits is sent on air for encrypted link as are sent on an unencrypted link.

5

APPLICATIONS

5.1 Introduction

The technology of Bluetooth has many applications in the everyday life. Personal computers, mobile phones, gaming, advertising are only some of the fields, where Bluetooth finds numerous applications and allow the users to enjoy its benefits.

5.2 Personal Computers

- Wireless connection of peripheral units like mouses, keyboards and printers.



Figure 5.1: Wireless keyboard and mouse.

- Wireless connection between laptops or desktops which are in a short distance from one another (in the same room) and the applications supported by them do not have major requirements in bandwidth.
- Connection between mp3 players or digital cameras with computers for the transfer of files.

5.3 Mobile Phones

- Wireless headsets for the creation of a wireless connection between the mobile phone and the receiver, so that the user can make or receive a call using voice command.



Figure 5.2: Wireless headsets.

- Mobile phones which can be connected with PDAs or computers or with other mobile phones for the transfer of music files, pictures or other types of data.



Figure 5.3: A PDA can be connected with a mobile phone via bluetooth.

5.4 Gaming

- Bluetooth application at the new consoles of Playstation 3 και Nintendo Revolution games, enabling wireless remote control which provides the players with great autonomy of movements.



Figure 5.4: Wireless remote control for the new gaming consoles.

5.5 Advertising



Figure 5.5: Bluetooth is also used in the field of advertising.

Some application examples are described below.

- **Mobile Marketing:** automatic sending of information, discount vouchers or offers to person standing in front or passing by your advertising sign, or even to those who are inside the point of sales. Make use of the “dead” waiting time of your prospective customers in order to inform them about your products, or make them an offer that will urge them to an immediate purchase.

- Mobile Games: transform a waiting room to a gaming room. Send branded games to your customers and visitors and give them the opportunity to win a prize, thus improving your company's profile and your contact with the customers.
- Mobile Entertainment: Offer your customers and visitors the opportunity to download free videos, MP3s and pictures, thus creating a fascinating entertaining environment.
- Internet access: Offer free internet access to your visitors who have a Bluetooth device. You will extend your visitors stay and see your sales increasing.
- Communicate interactively with your customers offering them information on demand. Organize contests and polls, or send them real-time alerts.

6

FUTURE WORK

6.1 Ultra-wideband (UWB)

Ultra-wideband (UWB) radio is a fast emerging technology with many unique attractive features that promotes major advances in wireless communications, networking, radar, imaging, and positioning systems. Research in UWB is still in its infancy stages, offering limited resources in handling the challenges facing the UWB communications. Understanding the unique properties and challenges of UWB communications as well as its application in competent signal processing techniques are vital in conquering the obstacles towards developing exciting UWB applications. UWB research and development has to cope with the challenges that limit their performance, capacity, throughput, network flexibility, implementation complexity, and cost.

Ultra-Wideband (UWB) is a technology for transmitting information spread over a large bandwidth (>500 MHz) that should, in theory and under the right circumstances, be able to share spectrum with other users. Regulatory settings of FCC are intended to provide an efficient use of scarce radio bandwidth while enabling both high data rate "[personal area network](#)" (PAN) wireless connectivity and longer-range, low data rate applications as well as radar and imaging systems. Ultra Wideband was traditionally accepted as [pulse radio](#), but the FCC and ITU-R now define UWB in terms of a transmission from an antenna for which the emitted signal bandwidth exceeds the lesser of 500 MHz or 20% of the center frequency. Thus, pulse-based systems—wherein each transmitted pulse instantaneously occupies the UWB bandwidth, or an aggregation of at least 500 MHz worth of narrow band carriers, for example in [orthogonal frequency-division multiplexing](#) (OFDM) fashion—can gain access to the UWB spectrum under the rules. Pulse repetition rates may be either low or very high. Pulse-based UWB radars and imaging systems tend to use low repetition rates,

typically in the range of 1 to 100 megapulses per second. On the other hand, communications systems favor high repetition rates, typically in the range of 1 to 2 giga-pulses per second, thus enabling short-range gigabit-per-second communications systems. Each pulse in a pulse-based UWB system occupies the entire UWB bandwidth, thus reaping the benefits of relative immunity to [multipath fading](#) (but not to [intersymbol interference](#)), unlike [carrier-based](#) systems that are subject to both deep fades and intersymbol interference.

A significant difference between traditional radio transmissions and UWB radio transmissions is that traditional systems transmit information by varying the power level, frequency, and/or phase of a sinusoidal wave. UWB transmissions transmit information by generating radio energy at specific time instants and occupying large bandwidth thus enabling a pulse-position or time-modulation. The information can also be imparted (modulated) on UWB signals (pulses) by encoding the polarity of the pulse, the amplitude of the pulse, and/or by using orthogonal pulses. UWB pulses can be sent sporadically at relatively low pulse rates to support time/position modulation, but can also be sent at rates up to the inverse of the UWB pulse bandwidth. Pulse-UWB systems have been demonstrated at channel pulse rates in excess of 1.3 giga-pulses per second using a continuous stream of UWB pulses (Continuous Pulse UWB or "[C-UWB](#)"), supporting forward error correction encoded data rates in excess of 675 Mbit/s. One of the valuable aspects of UWB radio technology is the ability for a UWB radio system to determine "time of flight" of the direct path of the radio transmission between the transmitter and receiver at various frequencies. This helps to overcome multi path propagation, as at least some of the frequencies pass on radio line of sight. With a cooperative symmetric two-way metering technique distances can be measured to high resolution as well as to high accuracy by compensating for local clock drifts and stochastic inaccuracies. Another valuable aspect of pulse-based UWB is that the pulses are very short in space (less than 60 cm for a 500 MHz wide pulse, less than 23 cm for a 1.3 GHz bandwidth pulse), so most signal reflections do not overlap the original pulse, and thus the traditional multipath fading of narrow band signals does not exist. However, there still is multipath propagation and inter-pulse interference for fast pulse systems which have to be mitigated by coding techniques.

6.2 Applications

Due to the extremely low emission levels currently allowed by regulatory agencies, UWB systems tend to be short-range and indoors applications. However, due to the short duration of the UWB pulses, it is easier to engineer extremely high data rates, and data rate can be readily traded for range by simply aggregating pulse energy per data bit using either simple integration or by coding techniques. Conventional OFDM technology can also be used subject to the minimum bandwidth requirement of the regulations. High data rate UWB can enable [wireless monitors](#), the efficient transfer of data from digital [camcorders](#), wireless [printing](#) of digital pictures from a camera without the need for an intervening [personal computer](#), and the transfer of [files](#) among [cell phone](#) handsets and other handheld devices like [personal digital audio and video players](#). UWB is used as a part of location systems and real time location systems. The precision capabilities combined with the very low power makes it ideal for certain radio frequency sensitive environments such as hospitals and healthcare. Another benefit of UWB is the short broadcast time which enables implementers of the technology to install orders of magnitude more transmitter tags in an environment relative to competitive technologies. UWB is also used in "see-through-the-wall" precision radar imaging technology, precision locating and tracking (using distance measurements between radios), and precision time-of-arrival-based localization approaches. It exhibits excellent efficiency with a [spatial capacity](#) of approximately 10^{13} bit/s/m².

Figure 6.1 illustrates an ultra wideband, handheld transceiver that was designed for full duplex voice and data transmission at rates of up to 128 kb/s (CVSD) and 115.2 kb/s (RS232). The radio has an operational center frequency in L-band (1.5 GHz) with an instantaneous bandwidth of 400 MHz (27% fractional BW). Peak power output from the UWB transceiver was measured at 2.0 Watts, with a resultant average power (worst case) of 640 mW. This results in a worst case power density of 1.6 pW/Hz. These units have a range of approximately 1 to 2 km (with small antennas shown and line-of-sight), and an extended range of 10 to 20 miles with small gain antennas.

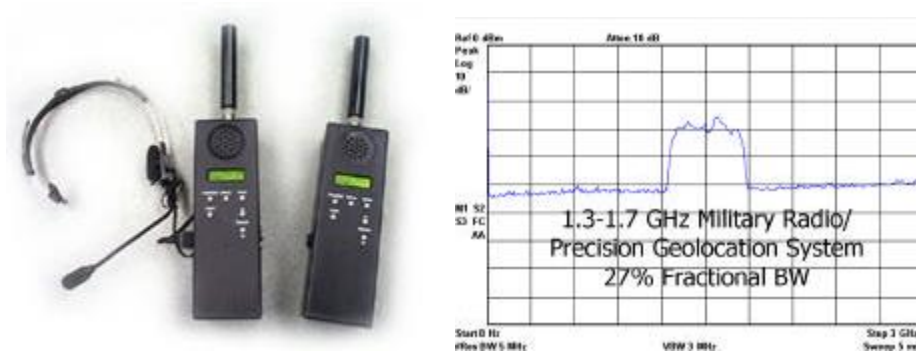


Figure 6.1: Full Duplex UWB Handheld Transceivers.

Figure 6.2 illustrates a rather unique UWB radio designed for non line-of-sight communications utilizing surface or ground wave propagation. To excite such propagation modes, the frequency of operation needs to be well below 100 MHz (e.g., Skolnik, 1990). Thus, this system was designed to operate in the frequency band from 30 to 50 MHz (50% fractional BW) and utilized a peak power output of approximately 35 Watts.

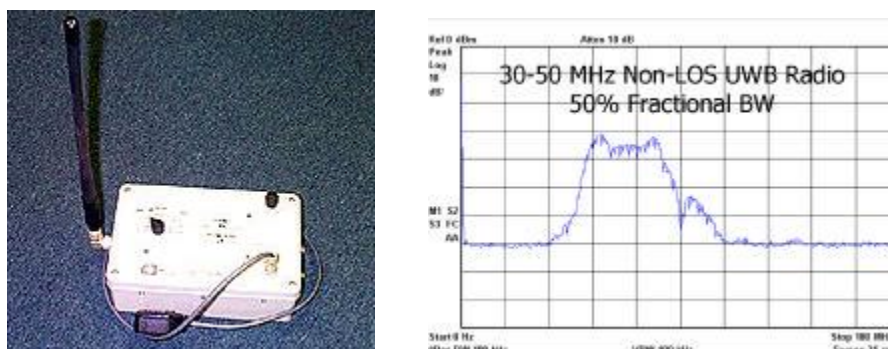


Figure 6.2: UWB Groundwave Communications System (w/measured power spectrum).

As in the above example, this radio was capable of both digital voice and data transmission to 128 kb/s and had an operational range over water of approximately 10 miles using a standard SINCGARS (30-88 MHz) VHF military antenna. Operational range over land depended upon terrain, but was observed to be from 1 to 5 miles with intervening foliage, buildings and hills. [Note that multipath cancellation is a serious problem over water, even with UWB impulse technology, because of the typically

low grazing angles (resulting in small differential delays between direct and reflected paths) and strong, out-of-phase reflection from the water surface. Thus, both higher frequency UWB systems as well as more conventional narrowband VHF/UHF radios were unable to provide the requisite performance.] Also unique to the receiver design was the addition of anti-jam circuitry to prevent loss of sensitivity due to strong, in-band interferers commonly observed in this frequency range. The units could also be operated in a "digipeater" mode in which packet store-and-forward techniques were used to transmit data from one radio to another via an intermediate repeater link.

Figure 6.3 illustrates a high-speed (up to 25 Mb/s) UWB radio designed for transmission of compressed video and command & control information across an asymmetric, bi-directional link. This system was also designed to operate in the 1.3 to 1.7 GHz region (27% fractional BW) with a 4W peak power output. An earlier design, developed under funding from the U.S. Defense Advanced research Projects Agency (DARPA), operated with a 500 MHz instantaneous bandwidth in the C-band region (5.4 to 5.9 GHz).



Figure 6.3: Asymmetric Video/Command & Control UWB Transceivers.

The ultimate goal of this design was to provide for up to 60 nautical mile, line-of-sight transmission to/from an unmanned aerial vehicle (UAV). A small parabolic dish antenna was used at the ground platform.

A variant of the above system is illustrated in operation in Figure 6.4 below. In this figure, a 2 Mb/s asymmetrical UWB link is used to transmit compressed video from a small unmanned ground vehicle (robot) through a UAV (unmanned helicopter) relay to a soldier ground station. The command & control signal (115.2 kb/s) to the robot is

relayed through the UAV; while compressed robot video transmissions (1-2 Mb/s) are relayed through the UAV to the soldier. Ranges to UAV and robot were a few kilometers.



Figure 6.4: Asymmetric Video/Command & Control UWB Transceivers.

Another unique application for UWB communications is illustrated by the tagging device shown below in Figure 6.5.



Figure 6.5: Ultra Wideband Tag & Tag Reader (*Vehicular Electronic Tagging and Alert System*).

This system, dubbed *Vehicular Electronic Tagging and Alert System (VETAS)*, was designed for the U.S. Department of Transportation to provide a means for keeping problem drivers (i.e., drivers who have repeated been convicted of traffic accidents or violations due to driving while under the influence of alcohol) off the road. The concept was to tag the vehicle with a device which relays a picture of the driver, together with information on the driver and the vehicle, to a roadside sensor in a police vehicle. The tag would be installed in lieu of impounding the vehicle or placing the convicted driver in jail.

Ultra wideband technology was considered for this application because of its ability to transmit large amounts of data at high speed in a mobile, multipath-prone environment. The UWB tag operated in the L-band region (1.4 to 1.65 GHz) and had a peak output power of approximately 250 mW for a demonstrated range of over 300 meters. The image of the driver was stored as a compressed JPEG file, together with additional ASCII data, in EEPROM and periodically transmitted at a 400 kb/s burst rate to a UWB receiver with display. The tag operated off of two AAA batteries (3.0V) and, in an operational scenario, was mounted behind the front grill of the automobile.

Figure 6.6 illustrates a set of prototype UWB transceivers designed for the U.S. Navy to provide a wireless intercom capability on-board Navy aircraft. The prototype UWB transceivers provide multichannel, full duplex, 32 kb/s digital voice over a range of approximately 100 meters. An ultra wideband waveform was selected because of its ability to operate in severe multipath (created by multiple RF reflections inside and around aircraft), and because of its non-interfering, low probability of intercept signature.



Figure 6.6: UWB *Wireless Intercom Communications System* (WICS).

Current intercommunications systems (ICS) designs for aircraft utilize lengthy, and often unwieldy, cords to physically attach the crewman's headset to a distributed audio (intercom) system. Such physical attachment presents a safety hazard to personnel, impedes movement throughout the platform and reduces mission effectiveness. Replacement of these mechanical tethers with wireless RF links is a desirable alternative.

Frequency of operation for the WICS transceivers was again in the L-band region (1.2 to 1.8 GHz). One of the unique features of the WICS design was the use of a

frequency division multiplex, time division multiple access (FDM/TDMA) strategy for full duplex, multi-user operation. Because of the extremely short duration pulsewidths and resulting low energy densities, UWB systems are much less vulnerable to intercept and ECM attack than conventional RF communications systems. As a consequence, they also minimize interference to other on-board electronics, such as sensitive flight control systems, GPS, etc. With an extremely low duty cycle, a very low power drain can be achieved, thereby providing communications capability for mission life exceeding 12 hours. The WICS program has recently received additional funding to further improve and miniaturize the design.

One of the most recent applications of UWB communications technology is to the development of highly mobile, multi-node, *ad hoc* wireless communications networks. Figure 6.7 illustrates such a system currently under development for the U.S. Department of Defense. The system is designed to provide a secure, low probability of intercept and detection, UWB *ad hoc* wireless network capability to support encrypted voice/data (to 128 kb/s) and high-speed video (1.544 Mb/s T1) transmissions.



Figure 6.7: UWB tactical *ad hoc* wireless network

[UWB handheld, network processing unit and applications computer]

A parallel effort, currently funded by the Office of Naval Research under a Dual Use Science and Technology (DUS&T) effort is developing a state-of-the-art, mobile *ad hoc* network (MANET) based upon an Internet Protocol (IP) suite to provide a connectionless, multihop, packet switching solution for survivable communications in a high link failure environment. The thrust of the DUS&T effort is toward commercialization of UWB technology for applications to high-speed (20+ Mb/s) wireless applications for the home and business.

A UWB application which bridges the gap between communications and radar is that of precision geolocation. Also see accompanying paper (Fontana, 2000). Figure 6.8, for example, illustrates a system designed to provide 3-dimensional location information utilizing a set of untethered UWB beacons and an untethered, mobile UWB rover. Precision location is derived from round trip, time-of-flight measurements using packet burst transmissions from the UWB rover and beacon transponders.



Figure 6.8: UWB Precision Geolocation System Transceiver.

The system in Figure 6.8 utilizes a 2.5 ns, 4 Watt peak, UWB pulse, again operating in the 1.3 to 1.7 GHz region. Line-of-sight range for the system is better than 2 kilometers utilizing small, omnidirectional vertically polarized (smaller) or circularly polarized (larger) antennas. Within a building, the range becomes limited by wall and obstacle attenuation; however, ranges exceeding 100 meters inside have been attained. A unique feature of the system is the ability to detect the pulse leading edge through the use of a charge sensitive, tunnel diode detector. Leading edge detection is critical to the resolution of the direct path from the plethora of multipath returns produced from internal reflections. The UWB geolocation system was originally developed to permit a soldier to determine his or her position to within 1 foot resolution in an urban environment. It is currently being used to augment a video capture system for 3-D modeling and for materiel location onboard a Navy ship.

Figure 6.9 illustrates an ultra wideband system designed as a precision altimeter and obstacle/collision avoidance sensor. Originally developed for the U.S. Marine Corps' *Hummingbird* unmanned aerial vehicle, the sensor has proved capable of detecting small diameter (0.25" or 6.35 mm) suspended wires to ranges beyond 250 feet. With a peak output power of only 0.2 Watts, the system operates in the C-band

region from 5.4 to 5.9 GHz (8.9% fractional BW) and has an average output power at 10 kpps of less than 4 μ W. Range resolution of the radar was better than one foot utilizing the leading edge detection capability.



Figure 6.9: *Hummingbird* UWB Altimeter and Collision Avoidance Sensor.

For the *Hummingbird* application, the system incorporated a linear forward-looking phased array (cf. Figure 9 right), and broad beamwidth side-looking antennas, for use in autonomous control. Interestingly, a predecessor of *Hummingbird* was developed for the U.S. Naval Air Systems Command as a multifunction precision altimeter, collision avoidance sensor and low data rate communications system. A 1 Watt version of the radar operated as a precision (1 foot resolution) radar altimeter to an altitude of better than 5000 feet.

Several variants of the *Hummingbird* radar have also been developed. For example, Figure 6.10 illustrates an ultra wideband backup sensor for the detection of personnel, vehicles and other objects behind large construction and mining vehicles.



Figure 6.10: UWB Backup Sensor.

Operating with approximately 250 mW peak in the C-band region from 5.4 to 5.9

GHz, the backup sensor utilizes a dual antenna configuration for the detection of objects as close as 1 foot to beyond 350 feet from the vehicle. Ultra wideband provides a significant advantage for this application because of the ability to provide precision range gating to eliminate clutter which, with conventional Doppler-based sensors, often results in large false alarm rates. This sensor was developed for the National Institute of Occupational Safety and Health.

Another variant of the *Hummingbird* collision avoidance sensor was developed as part of an electronic license plate for the U.S. National Academy of Sciences' Transportation Research Board (Figure 6.11 below). The UWB *Electronic License Plate* provides a dual function capability for both automobile collision avoidance and RF tagging for vehicle to roadside communications. Collision avoidance functions are achieved with a miniature, 500 MHz bandwidth C-band UWB radar; and RF tagging functions are accomplished with a low power, 250 MHz bandwidth L-band system.



Figure 6.11: UWB License Plate (Tag + Collision Avoidance Radar).

The UWB C-band radar utilized a 0.2W peak power (4 μ W average) waveform to achieve a range of better than 100 feet against other vehicles, with an accuracy of better than 1 foot. The L-band tag operated with a 0.3W peak power (500 μ W average) packet burst transmission to achieve a data throughput of 128 kb/s over a range exceeding 800 feet. An ultra wideband solution was chosen for the *Electronic*

License Plate because of its precision ranging capability (radar mode) and high multipath immunity (tag mode).

Another short range radar, this time operating in the X-band region of the spectrum, is shown below in Figure 6.12. This prototype sensor was developed for the U.S. Army Missile Command as a low probability of intercept and detection (LPI/D), anti-jam, radar proximity sensor for medium caliber, small caliber and submunition applications. The system exhibited an operational bandwidth of 2.5 GHz with a 10 GHz center frequency. Specifically designed for very short range applications (less than 6 feet), the UWB sensor has a 6 inch range resolution. With an average output power output of less than 85 nanowatts, a -4 dBsm target could be detected at a range of approximately 15 feet using small, microstrip patch antennas.

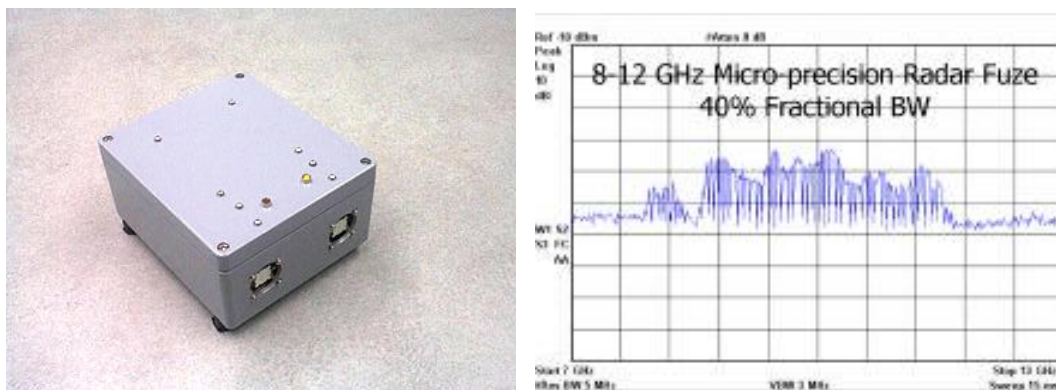


Figure 6.12: X-band UWB Fuze Sensor.

A variant of the X-band UWB radar fuse is currently being developed for DARPA's Micro Air Vehicle (MAV) program under a Phase II Small Business Innovation Research (SBIR) contract. Figure 6.13 illustrates a mockup of a 4 inch micro helicopter with an array of four X-band UWB antennas. Weight and size are obviously driving factors for this design, and a UWB chipset is being developed for an onboard collision and obstacle avoidance sensor.

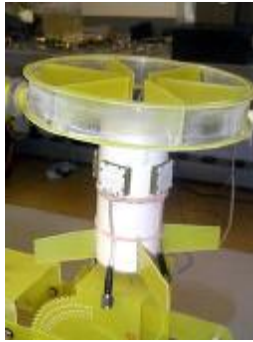


Figure 6.13: Mockup of 4" MAV with X-band UWB Antenna Array.

Figure 6.14 illustrates a UWB intrusion sensor radar which was designed for through the-wall penetration. With an L-band operational frequency and 33% fractional bandwidth, this system utilizes a 1 Watt peak UWB waveform to detect personnel through several intervening walls.



Figure 6.14: UWB Through-the-wall Intrusion Sensor.

Broad area surveillance coverage was provided for both in-building and outdoor field environments. An extended range system was also developed to detect and track human targets at distances exceeding 1000 feet. Figure 6.15 illustrates the switched antenna array used with this broad area surveillance system. Target azimuth and distance are determined and used to point a camera in the direction of the target.



Figure 6.15: UWB Antenna Array (*Spider*) for Extended Range Intrusion Sensor.

6.3 Bluetooth and Ultra-wideband (UWB)

- May 4, 2005

The Bluetooth Special Interest Group (SIG) announced its intent to work with the developers of the wireless technology commonly known as Ultra-wideband (UWB) to combine strengths of both technologies. This decision will allow Bluetooth technology to extend its long-term roadmap to meet the high-speed demands of synchronizing and transferring large amounts of data as well as enabling high quality video applications for portable devices. UWB will benefit from Bluetooth technology's manifested maturity, qualification program, brand equity and comprehensive application layer.

Bluetooth will still be important to maintain backward compatibility with existing devices on the market and future products not requiring the higher data rate. It has been apparent that members of the Bluetooth SIG would like to enable products with higher data rates. Joint development between Bluetooth technology and UWB is the fastest and most economical pathway for both technologies to meet the future demands of companies and end users. At the same time it is important to understand that Bluetooth is a global standard to a great extent driven by the adoption into mobile consumer devices like mobile phones, so not only is a requirement that world-wide regulation is achieved but also that it is done in a way so co-existence with future mobile standards is realized. The Bluetooth SIG's intention to employ UWB in their next generation products is a very positive step in allowing consumers to connect seamlessly between PCs, phones and consumer electronics equipment. This is an extremely positive move.

With this collaborative approach, it will be possible to maintain existing Bluetooth core values like low power, low cost and unique ad hoc connectivity while enabling future usage scenarios requiring higher data throughput. Such an example would be streaming high quality video between portable devices. As digital content size increases, the bit rate required to move data from device to device increases. As such, a classic Bluetooth usage scenario today of exchanging a file is more likely to require UWB speeds in the future. The collaboration of both groups is a natural and necessary evolution of the market. As consumers continue to increase the use of

portable and digital media devices, the need for standardized, higher performance, low power connectivity solutions becomes integral. Leveraging Bluetooth technology's established brand and traction in the consumer space with the higher data rate, lower power UWB technology should enable a faster time to market for next generation devices, and compliment the growing demand for connectivity.

UWB will benefit from Bluetooth technology's brand equity, market penetration and technical and organizational maturity. UWB can skip many time-intensive and costly hurdles in technology and market development by joining forces with a technology that is past that stage. Not only do companies want to leverage investments in Bluetooth technology, but 250 million consumers who also invested in Bluetooth technology will want those devices to work with future high data rate WPAN products.

- March 28, 2006

The Bluetooth SIG announced its selection of the WiMedia Alliance multiband orthogonal frequency division multiplexing (MB-OFDM) version of ultra-wideband (UWB) for integration with current Bluetooth wireless technology, thus taking the next step in its plan to create a version of the globally popular Bluetooth wireless technology with a high speed/high data rate option. This new version of Bluetooth technology will meet the high-speed demands of synchronizing and transferring large amounts of data as well as enabling high quality video and audio applications for portable devices, multi-media projectors and television sets. At the same time, Bluetooth technology will continue catering to the needs of very low power applications such as mice, keyboards and mono headsets, enabling devices to select the most appropriate physical radio for the application requirements, thereby offering the best of both worlds.

The WiMedia Alliance is a not-for-profit open industry association that promotes and enables the rapid adoption, regulation, standardization and multi-vendor interoperability of ultra-wideband (UWB) worldwide. The basis for the industry's first UWB standards (published by Ecma International), WiMedia UWB is optimized for wireless personal-area networks delivering high-speed (480Mbps and beyond), low-power multimedia capabilities for the PC, CE, mobile and automotive

market segments. Emphasizing peaceful coexistence with other wireless services, the WiMedia UWB common radio platform is designed to operate with application stacks developed by the 1394 Trade Association Wireless Working Group, the Certified Wireless USB Promoter Group and the Bluetooth SIG. WiMedia's board members include Alereon, HP, Intel, Kodak, Microsoft, Nokia, Philips, Samsung Electronics, Sony, STMicroelectronics, Staccato Communications, Texas Instruments and Wisair.

It is critical that the UWB technology be compatible with Bluetooth radios and maintain the core attributes of Bluetooth wireless technology – low power, low cost, ad-hoc networking, built-in security features, and ability to integrate into mobile devices. Backwards compatibility with the over 500 million Bluetooth devices currently on the market is also an important consideration. The Bluetooth SIG is satisfied that MB-OFDM UWB technology, offered by the WiMedia Alliance, is capable of meeting all of these requirements. The two organizations are dedicated to working together to ensure that the combined high-speed solution is optimized for mobile devices with very low power consumption.

One of the key components to the agreement between the Bluetooth SIG and the WiMedia Alliance will help UWB achieve global regulatory acceptance. Both parties have agreed to develop a high speed, high data rate Bluetooth solution that utilizes the unlicensed radio spectrum above 6 GHz. This move answers concerns voiced by regulatory bodies in both Europe and Asia.

The Bluetooth SIG Core Specification Working Group Charter and UWB Feature Requirements Document (FRD) have been approved by the Bluetooth SIG Board of Directors, signaling that work may commence. The requirements set by the UWB study group in the UWB FRD define what has to be done to create a solution appropriate for adoption by the Bluetooth SIG. Both groups will immediately begin work together on the specification draft within the Bluetooth SIG Core Specification Working Group. The Bluetooth SIG estimates this process to last approximately one year, with the first Bluetooth technology/UWB solution chip sets available for prototyping in Q2 2007.

- **January 7, 2008**

The Bluetooth SIG has brought together wireless technologies to create one wireless

option for consumers worldwide. The Bluetooth SIG is working with the WiMedia Alliance to use ultra-wideband (UWB) technology as the high speed channel for *Bluetooth* technology. The organization also welcomed Wibree technology into the *Bluetooth* wireless fold in 2007 and began work on an ultra low power *Bluetooth* specification. Both are expected in prototyping phase in 2008 with availability in the first half of 2009.

- April 22, 2009

From its annual All Hands Meeting in Tokyo this week, the [Bluetooth SIG](#) formally adopted [Bluetooth Core Specification Version 3.0 High Speed \(HS\)](#), or Bluetooth 3.0. This latest iteration of the popular short-range wireless technology fulfills the consumers' need for speed while providing the same wireless Bluetooth experience – faster. Manufacturers of consumer electronics and home entertainment devices can now build their products to send large amounts of video, music and photos between devices wirelessly at speeds consumers expect.

The UWB standard is firm and products are being sold, most of them as wireless USB connections. However, as the BT SIG's announcement says, "the Bluetooth SIG Ecosystem Committee determined that the previously established criteria for UWB Industry Acceptance would not be met prior to the final adoption of the Seattle (+HS) Core Release". Many designers will be disappointed that UWB was not chosen, but perhaps it is still a future possibility.

7

EXPERIMENT

7.1 Introduction

Bluetooth is a low power wireless standard for allowing a short range communications among multiple devices. It has become a global specification for such wireless connectivity. The technology of Bluetooth finds numerous applications in marketing and advertising fields. What makes this technology interesting and wide applicable is the fact that it is wide spread and allows the design of low power, small sized, and low cost radios that could be fit in many handheld devices.

In this paper we introduce our current work on exploring this area. We present an empirical investigation of user experience with Bluetooth-Based marketing and advertising activities. The main research goal of this study is to examine experimentally the attention of the user to his external environment, during an interactive communication by Bluetooth with his mobile phone, e.g. what happens when a user receives an advertising message by Bluetooth to his mobile phone while waiting for the bus or driving his car. Response time of the user to the received message or to the obstacles he faces to his external environment and how in general does a pushed content affect the user experience during an interaction via Bluetooth are investigated through an experimental procedure.

7.2 Equipment

7.2.1 Experimental Protocol Brief Description

Users will be presented with a mobile client application that will simulate the whole Bluetooth push content procedure. Programmable timers will be adjusted to mimic the scan times (long times for a crowded or full of obstacles environment

shorter for light traffic unobstructed settings) and push times (dependent mostly on file size) to fully simulate the user experience.

User attention span will be measured by response times and accuracy of response to questions and tasks relating to the content “pushed” to their mobile. They will be also tested according to their response to environmental stimuli (to simulate attention when walking to avoid obstacles or identifying the awaited bus). Environmental conditions will be simulated by an independent program projecting stimuli through a projector. User response will be measured by using a foot pedal.

7.2.2 PC Application

The external environment of the user is simulated in our experiment by an application, which runs on a personal computer. The data, which simulate the impulses of the external environment of the user, are presented through a projector and the user is asked to react to them by hitting a mouse, designed to be pressed by foot. We name this application `pc_appl`.

In the `pc_appl` the user sees on the screen shapes. One circle, which is still, and one square, which moves towards the circle. The user must react as fast as he can by hitting the foot-pedal (mouse), when the square becomes red. This procedure is called a cycle. We can repeat more cycles by changing the corresponding parameters to a special file of the application. In Figures 7.1 and 7.2 we can see two shots from the first cycle of the application.

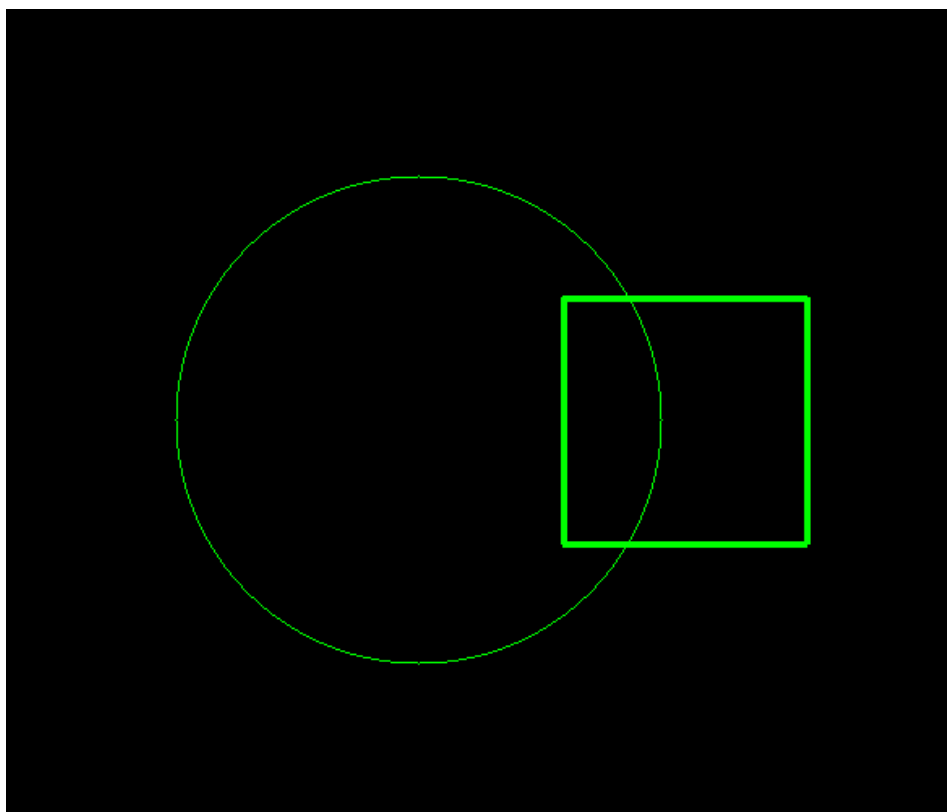


Figure 7.1: 1st cycle of the application.

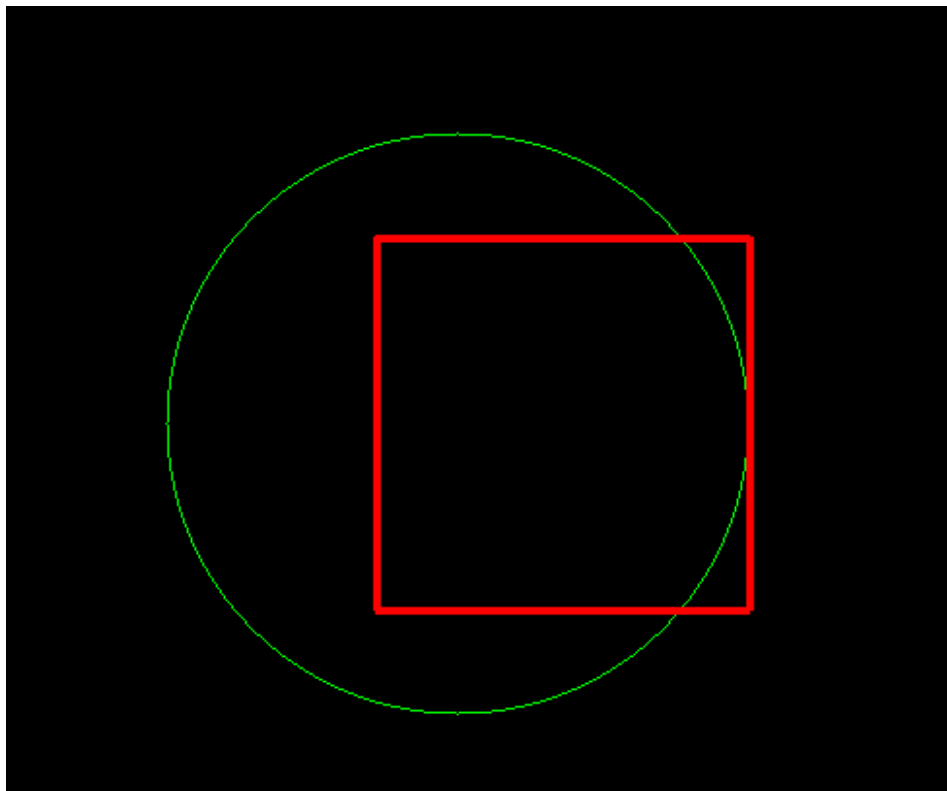


Figure 7.2: 1st cycle of the application. The user must react.

7.2.3 Mobile Application

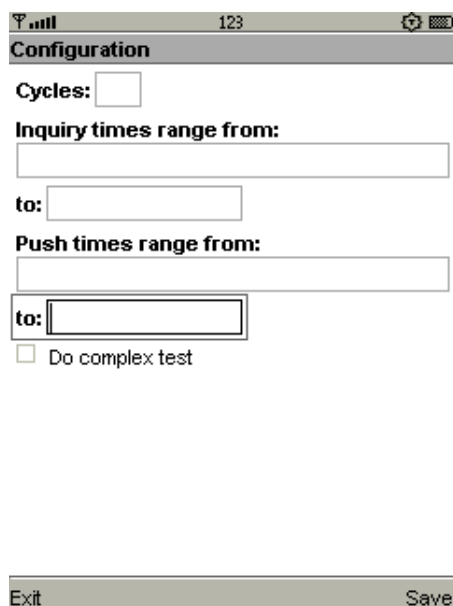
The interactive communication by Bluetooth is simulated by another application which runs to a mobile phone and does not use any Bluetooth technology, so that the results of the experiment have general acceptance. The application simulates hypothetical scan and push times and shows a message, which requires exact reactions from the user. We name this application mbl_appl.

We have 2 mobile applications with different difficulty level.

i) mbl_appl_A: The user must react correctly and as fast as he can to the instructions he receives with the message that appears. The messages require from the user to press a certain number.

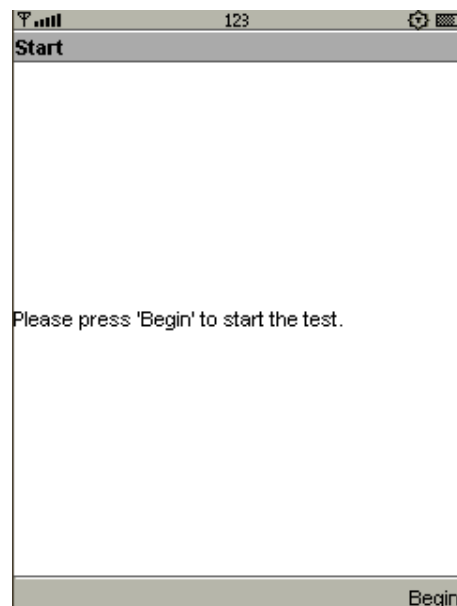
ii) mbl_appl_B: The user must react correctly and as fast as he can to the instructions he receives with the message that appears. The messages require from the user to execute a summation and press the last digit of the result.

In both applications we set the scan and push times in ranges and we also choose the number of cycles we want. The following Figures show the stages of the procedure.



The screenshot shows a mobile application interface with a status bar at the top displaying signal strength, the number 123, and battery level. The main screen is titled "Configuration". It contains several input fields: "Cycles:" followed by a small square input field; "Inquiry times range from:" followed by a long horizontal input field; "to:" followed by a shorter horizontal input field; "Push times range from:" followed by another long horizontal input field; and "to:" followed by a shorter horizontal input field. Below these fields is a checkbox labeled "Do complex test". At the bottom of the screen, there are two buttons: "Exit" on the left and "Save" on the right.

Figure 7.3: We set the number of cycles, the range of scan and push times and if we want mbl_applA or mbl_appl_B.



The screenshot shows a mobile application interface with a status bar at the top displaying signal strength, the number 123, and battery level. The main screen is titled "Start". It contains a single line of text: "Please press 'Begin' to start the test." At the bottom of the screen, there is a single button labeled "Begin".

Figure 7.4: The user presses “Begin” to start the application.



Figure 7.5: The application simulates hypothetical scan time.

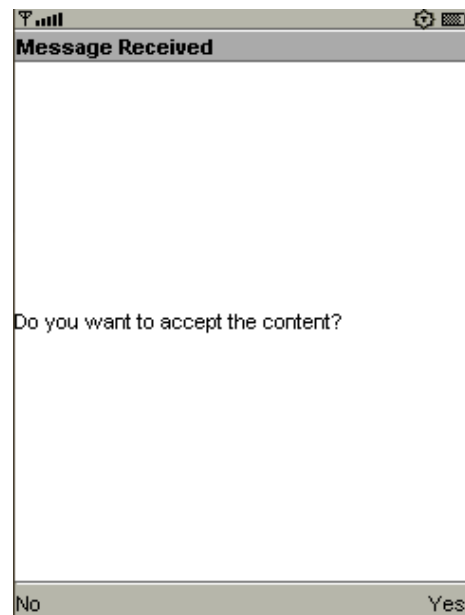


Figure 7.6: The user is asked if he the wants to accept a pushed content.

If the user accepts the content, the application goes on.

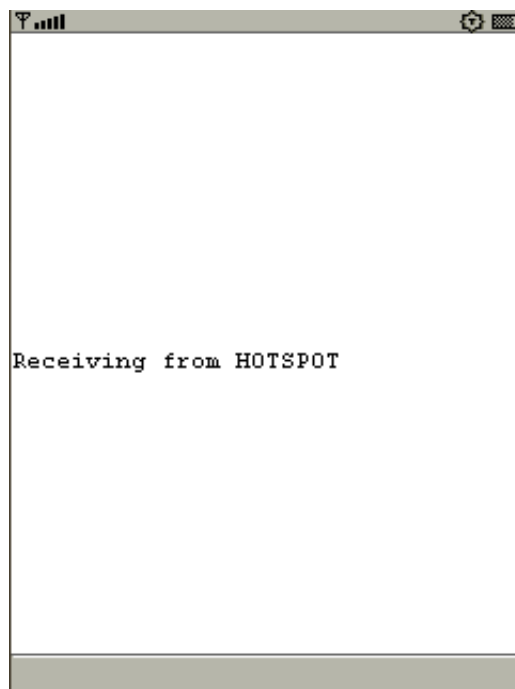


Figure 7.7: The application simulates the hypothetical push time.

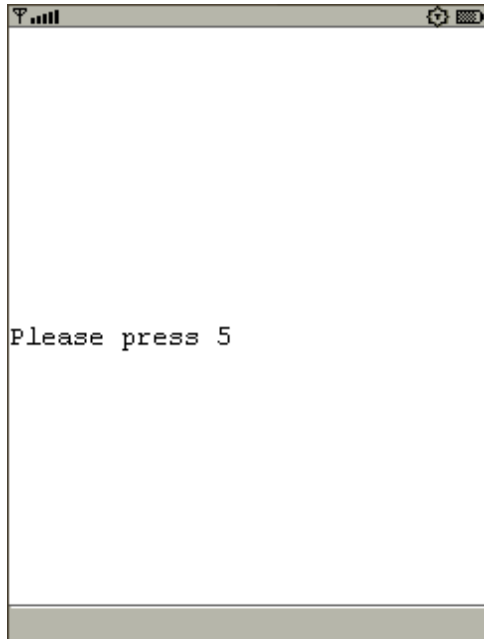


Figure 7.8: The user receives the content and must react correctly. (mbl_appl_A)

OR

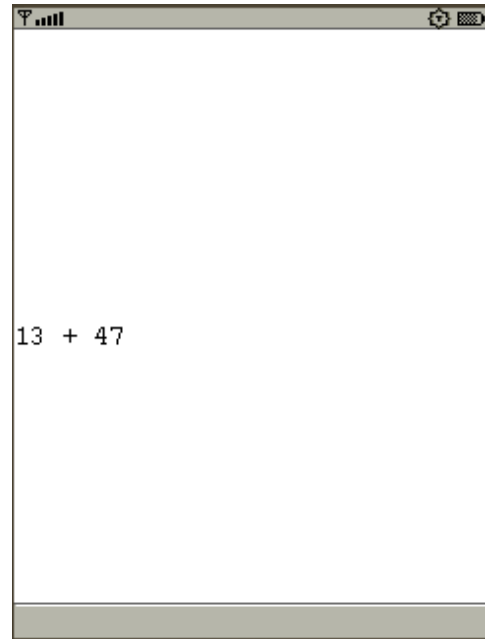


Figure 7.9: The user receives the content and must react correctly. (mbl_appl_B)

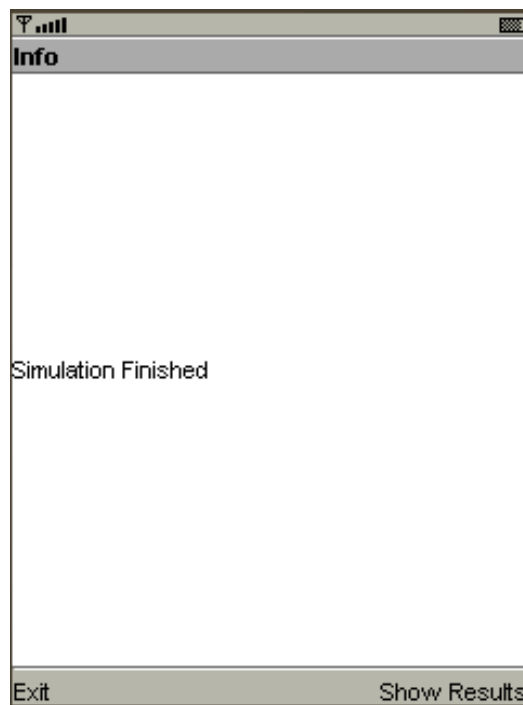


Figure 7.10: The application is finished.

If the user rejects the content, he does not receive anything.

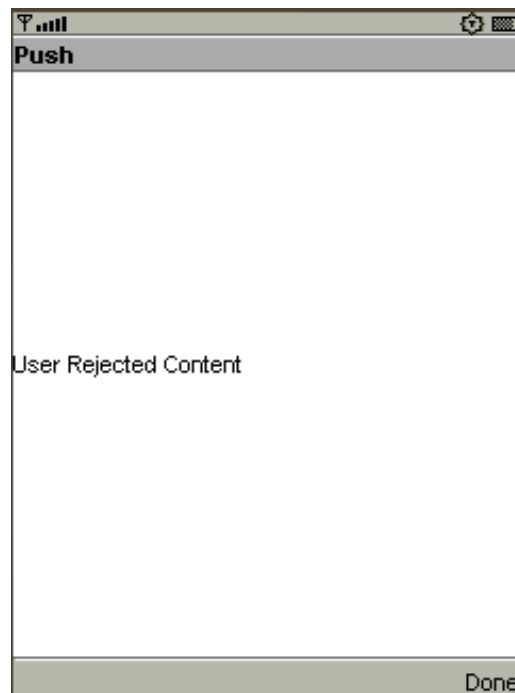


Figure 7.11: User rejected content.

The users were asked to accept always the content.

7.3 Experimental Procedure

The participants executed an experimental procedure. They held a mobile phone, in which run the mbl_appl (first A and then B). They were asked to stand in front of a wall, where the data from the pc_appl were presented through the projector. The two applications were synchronized.

We make the following separation of cases:

1st case : shortSCAN-shortPUSH

2nd case : longSCAN-shortPUSH

3rd case : shortSCAN-longPUSH

4th case : longSCAN-longPUSH

The values range for shortSCAN, longSCAN, shortPUSH, longPUSH is specified experimentally in section 7.5.

By organizing our experiment this way, we have 8 different missions: 4 with mbl_appl.A and 4 with mbl_appl.B. For each of these missions we repeated 10 cycles. In the experiment participated 40 users.

7.4 Related Work

7.4.1 Scan time

SCAN time is the time that a Bluetooth hotspot needs in order to discover all nearby Bluetooth devices. This time depends on various factors, such as the number of devices, the distance or the conditions of the environment (obstacles). We briefly present parts from other scientific studies and empirical researches, which will help us to estimate the appropriate SCAN times for our experimental procedure.

i) Number of Bluetooth devices

In the simulated scenario, we considered a single piconet in which only one device always assumes the role of master, while the others assume the role of slaves. In particular, slaves are always in inquiry scan mode and they start listening on frequencies from the master. Since discovery time depends on the number of slaves in the piconet, we performed our simulations varying this number in the range.

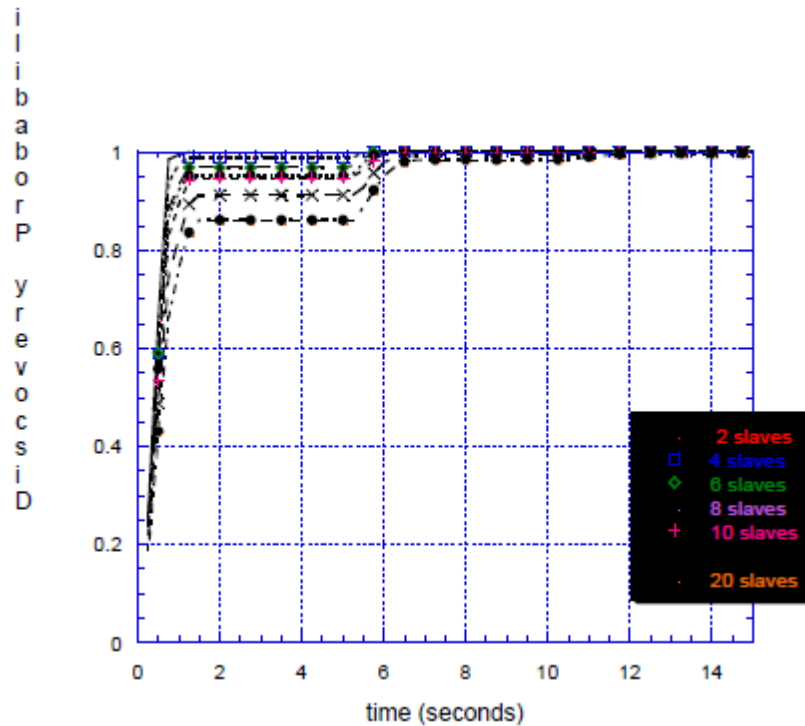


Figure 7.12: Inquiry and connection management.

ii) Distance and environment

The measurements were taken in a laboratory environment. The distance between the Bluetooth Sensor and the mobile phone ranged from 0.4m to 28 meters, so that in the case of 0.4 m the devices were placed on top of a desk, otherwise the measurements were carried out in a 28-meter corridor. There were also other Bluetooth devices present, and the environment has a WLAN network installed. It is well known that Bluetooth and WLAN networks that are operating in the same 2.4 GHz ISM (industrial, scientific, medical) band interfere with each other, especially if the devices are close to each other. It may take some extra time for the Bluetooth protocol to cope with the interference.

Distance [m]	<i>n</i>	Mean [s]	Std dev [s]	Median [s]	Min [s]	Max [s]
0.4	20	18.7	8.69	17.5	6.0	42.5
5	10	18.2	9.50	18.2	6.5	37.8
8	10	22.3	10.92	21.5	3.5	38.7
10	10	27.5	14.23	28.8	4.6	41.7
12	10	33.0	21.94	21.8	15.9	77.1
14	10	22.8	8.77	19.3	14.1	38.4
16	10	18.2	8.38	18.5	3.9	35.7
18	10	19.2	4.64	19.6	10.7	24.8
20	10	26.4	15.69	20.4	5.4	50.9
22	10	33.2	22.75	33.2	6.9	77.8
24	10	30.2	13.32	30.6	12.6	49.0
26	9	31.4	15.90	25.8	15.9	61.9
28	10	36.3	29.37	32.0	5.7	107.7
All	139	25.4	15.86	19.6	3.5	107.7

Figure 7.13: Summary of positioning time measurements at different distances in a laboratory environment.

Previous investigations into Bluetooth links have suggested 2 seconds as a typical setup time between two unknown devices, although some experiments have shown much higher values of 10s and greater (for unknown reasons). A quick read of the Bluetooth specification also indicates that 2s is a reasonable number, and that 10s should be an absolute maximum.

In Figure 7.13 the values are much higher, but in this case we have many other Bluetooth devices present and a WLAN network installed, working at the same time.

7.4.2 Push time

PUSH time is the time that is needed until the pushed content is downloaded to the user's mobile phone. This time depends on the filesize of the pushed content and the transfer rate of Bluetooth. The abstract below provides some relevant experimental data.

Push message sending latencies were also measured by sending a message to the Push Proxy Gateway of the Octopus network and measuring the time it took until the

SI message consisting of two SMS messages arrived at the mobile handset. This latency is highly dependent on the GSM network and its SMS throughput. Most of the time is spent paging the device, i.e. locating it on the network, and sending the binary-encoded SI SMS messages to it. The average sending latency measured over 15 measurements was 11.6 seconds (std. dev. 0.41 s, min 10.9 s, max 12.2 s). [2]



Figure 7.14: Example of a pushed content.

7.5 Scan and Push Times

7.5.1 Scan time

In order to estimate the values range of this time we executed an experiment, so that the choice of the values range for shortSCAN and longSCAN is based on experimental data.

Experiment: A Bluetooth hotspot repeated constantly 10 times the inquiry mode in order to discover nearby mobile devices. We measured the SCAN time needed in each case.

i) Number of devices: 3

Distance: close to the hotspot

Obstacles: No

	<u>SCAN time (ms)</u> Device1	<u>SCAN time (ms)</u> Device2	<u>SCAN time (ms)</u> Device3	
	6019	4912	9223	
	8148	5618	10114	
	7887	9442	6927	
	5656	7611	12315	
	4123	7952	7618	
	8423	5111	15321	
	7111	5505	13012	
	6924	3108	8764	
	9441	8715	10082	
	8348	6532	11025	
Average:	7208	6450	10440	8033

Table 7.1: 3 devices, small distance, no obstacles.

ii) Number of devices: 3

Distance: far from the hotspot

Obstacles: Yes

	<u>SCAN time (ms)</u> Device1	<u>SCAN time (ms)</u> Device2	<u>SCAN time (ms)</u> Device3	
	9911	12318	16012	
	15012	11059	12118	
	13381	14326	13231	
	12652	10002	14001	
	11183	11481	14226	
	8022	13437	15632	
	9154	15773	18012	
	13915	13061	12115	
	11027	9889	13331	
	10341	10638	13225	
Average:	11459	12198	14190	12616

Table 7.2: 3 devices, long distance, obstacles.

iii) Number of devices: 10

Distance: close to the hotspot

Obstacles: No

	<u>SCAN time (ms)</u> Device1	<u>SCAN time (ms)</u> Device2	<u>SCAN time (ms)</u> Device3	
	4100	4100	53514	
	13227	18256	53565	
	10149	44300	53250	
	7148	29237	29237	
	14373	19253	30619	
	7397	5039	31249	
	14420	14420	16960	
	8512	14997	8512	
	15835	48168	26273	
	7865	31643	36008	
Average:	10303	22081	33919	22101

Table 7.3: 10 devices, small distance, no obstacles.

(*In this case we chose randomly 3 out of 10 devices.)

iv) Number of devices: 10

Distance: far from the hotspot

Obstacles: Yes

	<u>SCAN time (ms)</u> Device1	<u>SCAN time (ms)</u> Device2	<u>SCAN time (ms)</u> Device3	
	18147	29038	28147	
	30182	15482	22217	
	28087	28087	29888	
	14530	48769	36486	
	24306	38819	38819	
	26591	36703	21129	
	16452	11291	16291	
	21404	15104	25104	
	41356	24214	23214	
	35788	30388	41916	
Average:	25684	27789	28321	27265

Table 7.4: 10 devices, long distance, obstacles.

(*In this case we chose randomly 3 out of 10 devices.)

Different Cases	Average SCAN time	Conclusions
i) Number of devices: 3 Distance: close to the hotspot Obstacles: No	8033sec	shortSCAN: 5-15sec
ii) Number of devices: 3 Distance: far from the hotspot Obstacles: Yes	12616sec	
iii) Number of devices: 10 Distance: close to the hotspot Obstacles: No	22101sec	longSCAN: 20-30sec
iv) Number of devices: 10 Distance: far from the hotspot Obstacles: Yes	27265sec	

7.5.2 Push time

The theoretical Bluetooth transfer rate is 1Mbit/s, however practically it varies between 30KB/s and 90KB/s. That means that in the best case scenario a device can receive 500KB in 5.5 seconds and in the worst in ~18 sec. A normal file size for a pushed content is approximately 250-300KB. We make the following separation of cases:

i) Filesize: 250-300KB

Transfer rate: 30-90KB/s

Values range: 4-12sec

ii) Filesize: 750-800KB

Transfer rate: 30-90KB/s

Values range: 15-23sec

Different Cases	Conclusions
i) Filesize: 250-300KB Transfer rate: 30-90KB/s	shortPUSH: 4-12sec
ii) Filesize: 750-800KB Transfer rate: 30-90KB/s	longPUSH: 15-23sec

7.6 Measurements

In our experiment we took three measurements:

1. The response time of the user to the instructions he receives from the message of the mbl-appl.
2. How correctly did he execute the instructions.
3. How many times did he miss to press the foot-pedal when he should, which means obstacles of the environment he did not avoid.

In order to measure the reaction of the user as regards the instruction he receives and also the obstacles he has to avoid, we use a penalty points system. When the user misses to execute correctly or misses to execute at all the instruction he receives from the message of the mbl-appl, he is charged with 1 penalty point. When the user misses to press the foot-pedal when he should, he is also charged with 1 penalty point.

7.6.1 Response Time Mobile

		<i>mbl_appl_A(simple)</i>		
		<i>response_time_mbl (msec)</i>		
<i>User</i>	SCAN:5-15sec PUSH: 4-12sec	SCAN:5-15sec PUSH:15-23sec	SCAN:20-30sec PUSH:4-12sec	SCAN:20-30sec PUSH:15-23sec
1	2155	2796	2529	3335
2	2157	1959	2110	2619
3	1982	2770	2417	4269
4	2000	2125	2177	3452
5	2396	3271	3065	3226
6	2634	2936	2686	4424
7	2318	2770	2677	3084
8	2859	3259	3129	3495
9	2420	2904	2658	3803
10	1822	2810	1828	3083
11	2412	3173	3126	3414
12	2729	3048	3147	4990

13	2078	3809	3231	4047
14	2582	4032	3606	4553
15	2190	2606	2588	2773
16	1645	2088	1758	2990
17	3967	5115	4117	5586
18	2398	4027	4024	5162
19	1841	2138	1984	3829
20	2245	3401	3263	4720
21	1883	3583	2699	3773
22	1540	1999	1453	2120
23	1901	2525	2413	2647
24	2722	2958	2702	4706
25	2924	3627	3203	4198
26	3218	3862	3206	4221
27	1972	2180	2024	2770
28	3132	4581	3606	5248
29	2951	3560	3063	3623
30	1816	2238	1818	2510
31	2228	3989	2579	4312
32	2191	2381	2254	3295
33	1702	2865	2344	2904
34	3562	4490	3604	5164
35	3594	4442	3822	5275
36	1649	2805	1980	3426
37	2999	4156	3327	4458
38	2684	2957	2700	4830
39	2431	3161	2924	3673
40	1487	2509	2364	3052
Average	2385,40	3147,63	2755,13	3826,48

Table 7.5: Response_time_mbl (msec), mbl_appl_A (simple).

<i>mbl_appl_B(complex)</i>				
<i>response_time_mbl (msec)</i>				
<i>User</i>	SCAN:5-15sec PUSH: 4-12sec	SCAN:5-15sec PUSH:15-23sec	SCAN:20-30sec PUSH:4-12sec	SCAN:20-30sec PUSH:15-23sec
1	3147	3591	2899	3806
2	3080	3243	4649	4762
3	3251	3645	3066	4495
4	2656	3695	3173	4152
5	2817	3996	3330	3933
6	3042	5839	2914	4819
7	2458	2730	2668	3522

8	3202	4527	2718	4868
9	2553	3219	3247	4541
10	2177	2916	2790	4447
11	3844	3850	3758	4385
12	3368	4305	4003	5069
13	3001	4011	3331	4452
14	3609	4447	3849	4811
15	2528	4037	3506	4525
16	2578	2689	2636	3089
17	4245	5137	4961	6789
18	3964	4242	4235	5385
19	4177	4201	4403	4449
20	2968	4908	4547	5513
21	2457	3669	3311	3844
22	2844	3385	2853	3432
23	3166	4205	4072	5136
24	3253	4042	3494	4989
25	3121	3934	3820	4434
26	3429	4137	3523	4415
27	2753	3332	3201	3986
28	3576	5047	3807	5848
29	4272	3741	3253	4559
30	3058	3440	3365	3845
31	2622	4296	3798	4785
32	3204	3704	3365	4245
33	2114	3036	2894	4376
34	3905	4531	4496	5830
35	4282	4461	4918	5342
36	2428	3434	2739	3892
37	3752	4543	3982	5074
38	3230	3851	3831	5582
39	3916	4460	4413	4698
40	2602	3050	2954	5005
Average	3166,23	3938,15	3569,30	4628,23

Table 7.6: Response_time_mbl (msec), mbl_appl_B (complex).

7.6.2 Penalty Points Mobile

		<i>mbi_appl_A(simple)</i>		
		<i>penalty_points_mbl</i>		
<i>User</i>	SCAN:5-15sec PUSH: 4-12sec	SCAN:5-15sec PUSH:15-23sec	SCAN:20-30sec PUSH:4-12sec	SCAN:20-30sec PUSH:15-23sec
1	0	0	0	1
2	0	0	3	1
3	0	0	0	0
4	0	0	0	0
5	0	0	0	1
6	1	0	0	0
7	0	0	0	1
8	0	0	0	0
9	2	2	1	1
10	0	0	0	1
11	0	1	0	2
12	0	0	0	0
13	0	2	1	1
14	0	0	0	0
15	0	1	0	1
16	0	0	0	4
17	0	1	0	1
18	0	0	0	1
19	0	0	0	0
20	0	0	0	2
21	0	0	0	0
22	0	0	0	1
23	0	0	0	1
24	0	0	0	0
25	0	1	0	1
26	0	2	0	2
27	0	0	0	0
28	0	0	0	0
29	0	2	0	2
30	0	0	0	1
31	0	1	1	1
32	0	1	0	3
33	0	0	0	0
34	0	2	1	2
35	0	1	1	1
36	0	2	1	2
37	1	2	1	3

38	0	0	0	0
39	0	0	0	0
40	0	2	1	2
Average	0,10	0,57	0,28	1

Table 7.7: Penalty_points_mbl, mbl_appl_A (simple).

		<i>mbl_appl_B(complex)</i>		
		<i>penalty_points_mbl</i>		
<i>User</i>	SCAN:5-15sec PUSH: 4-12sec	SCAN:5-15sec PUSH:15-23sec	SCAN:20-30sec PUSH:4-12sec	SCAN:20-30sec PUSH:15-23sec
1	0	1	0	1
2	0	1	0	1
3	0	0	1	0
4	0	1	0	1
5	1	1	0	0
6	1	1	1	1
7	0	0	0	2
8	0	0	1	0
9	1	2	2	2
10	1	1	2	1
11	0	2	1	1
12	0	0	0	0
13	1	3	1	2
14	0	1	0	0
15	0	0	0	1
16	0	0	0	2
17	0	0	0	0
18	0	0	0	1
19	0	0	0	0
20	0	2	0	3
21	2	3	2	3
22	0	0	0	2
23	2	3	3	3
24	0	0	0	0
25	1	2	1	2
26	1	3	1	4
27	1	2	1	3
28	1	1	1	2
29	0	0	0	3
30	1	2	1	3
31	1	2	1	3
32	0	3	2	4

33	0	0	0	1
34	1	1	1	3
35	1	2	0	2
36	2	3	2	4
37	1	2	1	3
38	1	1	1	2
39	0	0	1	1
40	1	2	1	4
Average	0,55	1,20	0,72	1,78

Table 7.8: Penalty_points_mbl, mbl_appl_B (complex).

7.6.3 Penalty Points PC

		<i>mbl_appl_A(simple)</i>		
		<i>penalty_points_pc (%)</i>		
<i>User</i>	SCAN:5-15sec PUSH: 4-12sec	SCAN:5-15sec PUSH:15-23sec	SCAN:20-30sec PUSH:4-12sec	SCAN:20-30sec PUSH:15-23sec
1	20,37	26,44	24,74	28,79
2	9,23	10,1	9,82	13,74
3	14,29	17,28	17,35	27,91
4	15,52	18,52	17,19	20,77
5	14,75	24,42	19,23	32,5
6	19,3	29,41	24,3	31,39
7	16,07	20,93	17,78	21,32
8	26,76	27,55	26,47	29,01
9	13,24	23,4	22,33	25
10	31,75	18,56	21,65	19,38
11	29,85	36,84	31,73	38,17
12	18,75	21,74	20	23,88
13	17,19	21,43	21,37	23,36
14	20,59	34,78	29,46	37,24
15	5,36	11,22	9,9	14,49
16	4,35	5,66	4,17	6,06
17	11,94	20	12,5	20,71
18	32	38,68	31,63	45,65
19	10,94	23,91	14,29	25,36
20	15,38	16,48	15,53	18,85
21	18,84	23,47	20,75	25,56
22	14,06	18,18	18,1	21,83
23	22,45	26,97	24,75	31,65
24	14,08	17,78	16,04	19,38
25	17,65	22,34	19,8	23,31

26	23,61	26,38	24,04	28,36
27	21,82	25,88	24,47	27,27
28	14,08	21,74	18,37	23,13
29	14,06	17,39	16,19	20,59
30	16,4	19,57	17,82	21,17
31	15,25	17,35	16,35	18,31
32	12,07	15,12	15,38	16,91
33	15,38	20,65	17,35	21,01
34	20,51	21,92	20,95	27,94
35	17,91	21,18	19,66	23,66
36	21,43	21,65	21,01	25,37
37	24,62	26,73	25,69	28,57
38	15,63	16,67	15,46	17,16
39	16,36	17,14	17,39	19,72
40	15,38	16,3	16,16	17,16
Average	17,4805	21,5440	19,4293	24,0410

Table 7.9: Penalty_points_pc (%), mbl_appl_A (simple).

<i>mbi_appl_B(complex)</i>				
<i>penalty_points_pc (%)</i>				
<i>User</i>	SCAN: 5-15sec PUSH: 4-12sec	SCAN:5-15sec PUSH:15-23sec	SCAN:20-30sec PUSH:4-12sec	SCAN:20-30sec PUSH:15-23sec
1	22,58	26,37	23,76	28,36
2	8,93	11,36	10,81	14,39
3	10	11,7	12,77	18,66
4	23,21	26,44	25	32,56
5	22,58	29,35	31,63	32,85
6	25,81	28,85	28,97	30,83
7	21,43	25,81	23,16	29,23
8	29,33	30,34	29,59	31,54
9	14,75	17,31	16,98	27,54
10	18,97	18,68	17,24	20,29
11	33,33	40	35,04	43,88
12	24,62	26,8	25,44	30,43
13	21,43	23,91	22,12	26,09
14	19,12	23,76	18,37	24,63
15	11,43	12,77	12,73	22,86
16	7,69	12,5	11,32	18,66
17	11,11	12,15	10,48	12,86
18	35,94	37,18	36,94	38,41
19	13,24	13,4	12,73	14,29
20	15,63	18,75	12,5	20,9
21	21,74	25,51	23,53	28,97

22	16,07	19,1	18,37	22,63
23	31,34	36,17	34,95	37,21
24	17,24	20,65	18,45	21,38
25	20,29	23,91	21,43	29,29
26	24,62	27,37	25,9	30,43
27	24,14	26,37	25,24	29,01
28	17,19	21,43	19,81	32,85
29	19,35	22,11	20,19	24,11
30	18,75	21,9	20,54	25,55
31	18,33	18,68	17,7	20,61
32	14,29	17,2	16,83	19,31
33	19,67	21,11	20,39	23,13
34	23,88	25,81	24,53	28,89
35	19,05	21,98	21,3	24,24
36	22,06	24,69	23,71	26,43
37	26,92	27,37	26,67	30,28
38	15,25	15,56	15,46	17,16
39	17,46	19,19	18,8	22,3
40	18,46	18,63	18,75	20,69
Average	19,9308	22,5543	21,2533	25,8433

Table 7.10: Penalty_points_pc (%), mbl_appl_B (complex).

8

STATISTICAL ANALYSIS

8.1 Analysis of Variance (ANOVA)

ANOVA is a statistical method which will help us to answer the following question: Are the four different groups of numbers each time varying in a systematic manner?

8.1.1 Between- and Within-Groups Variation

The variation between the groups represents systematic variation due to the effect we are interested in (duration of scan and push time combination). The variation within the groups represents variation due to chance, that is, random variation that is not due to the effect that we are investigating. The between-groups variation and the within-groups variation for any groups can be mathematically determined. In formal statistical analysis, the word *variation* is replaced by the term *variance*.

The between-groups variance is often called the effect variance and the within-groups variance is often called the error variance (the error variance is due to chance). If the effect variance is large compared to the error variance, the groups are more likely to be defined as differing reliably. If the effect variance is small relative to the error variance, this is not so.

A good way to compare any two quantities is to form a ratio. Consider this ratio:

$$\frac{\text{Effect Variance (Between-Groups Variance)}}{\text{Error Variance (Within-Groups Variance)}}$$

The larger the value produced by this ratio, the more likely groups are to be truly different and therefore to be defined by statistical analysis as different.

8.1.2 Level of Significance

Statistical analysis will tell us whether the groups differ significantly or not. That is, it will tell us whether the effect size, the difference between the group means, is significant. In statistics the word *significant* has a precise meaning. If a result is statistically significant, it tells us that the group means are too different to have been that way by chance alone. In other words, the effect under investigation had a significant influence. The level of significance, $p < 0.05$, is typically employed in statistics. If a particular result is judged likely to occur less than 5 times out of 100 by chance alone, this result is said to be significant at the $p < 0.05$ level of significance. This means that the probability of getting that result by chance alone was less than 5%.

8.1.3 F-test

The statistical test used in ANOVA is called the F-test, after the statistician R.A. Fisher who developed it. Therefore,

$$F = \frac{\text{Effect Variance}}{\text{Error Variance}}$$

As we previously mentioned, the larger the value of F is, the more likely groups are to be truly different and therefore to be defined by statistical analysis as different. The condition of F-test and ANOVA is that the sample is normal.

In our statistical analysis we used the method of Repeated-Measures Analysis of Variance and a statistical program called SPSS 12.0 for Windows. All the measurements were examined as regards their normality, before ANOVA was applied, with the way in the example below:

Tests of Normality						
	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
group_1	,149	10	,200*	,956	10	,739

*. This is a lower bound of the true significance.

a. Lilliefors Significance Correction

Table 8.2 : Example of normality check.

If p values for both Kolmogorov-Smirnov test and Shapiro-Wilk test are over 0,05, the sample is normal. ($p = 0,200 > 0,05$ and $p = 0,739 > 0,05$). We made this procedure for all groups of measurements.

8.2 Response Time Mobile

8.2.1 Mobile Application A (simple)

8.2.1.1 Scan and Push Times Graphs

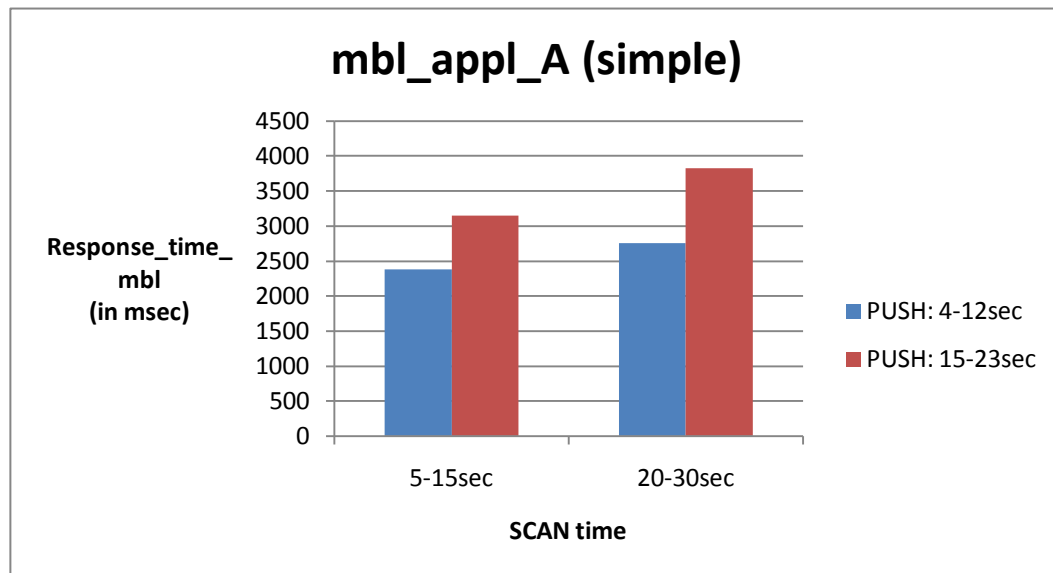


Figure 8.1: Response_time_mbl, mbl_appl_A (simple)~Scan time .

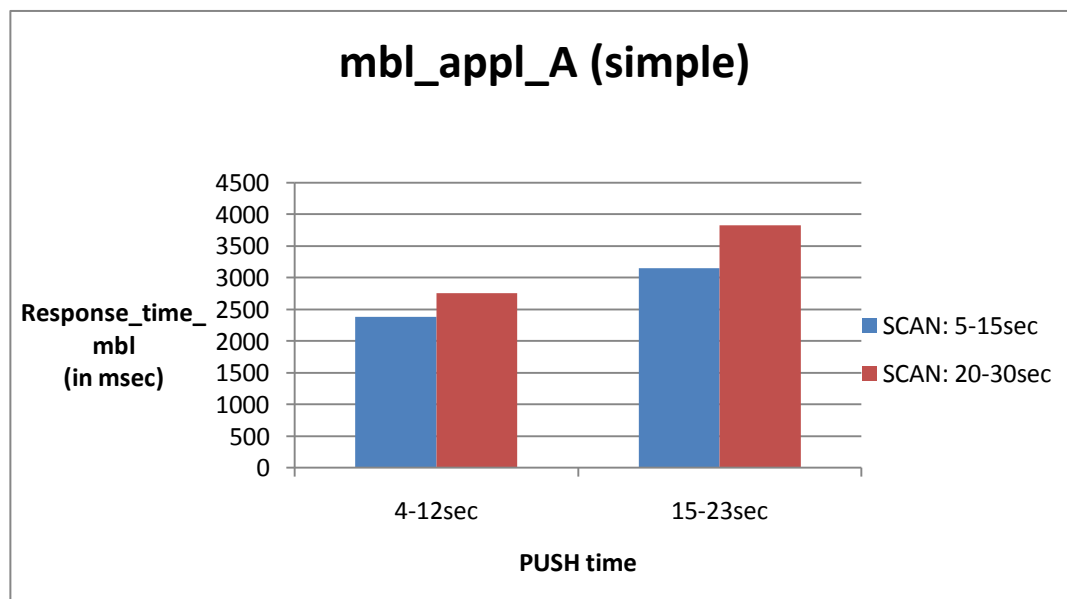


Figure 8.2: Response_time_mbl, mbl_appl_A (simple)~Push time .

From Figures 8.1, 8.2 we can see that in the first mobile application, which is the simple one, longer scan and push times mean longer response time. When the mobile phone needs too long time to be discovered and then too long time to receive the pushed content, the user is not concentrated on his phone.

8.2.1.2 ANOVA

In this section we applied the statistical method ANOVA in order to examine if and how the response time of each user is affected by the combination of scan and push times, by the gender or by the age.

Within-Subjects Factors

Measure: MEASURE_1

sc pu	Dependent Variable
1	shortSCAN_ shortPUSH
2	shortSCAN_ longPUSH
3	longSCAN_ shortPUSH
4	longSCAN_ longPUSH

Table 8.2: The four combinations of scan and push times.

Between-Subjects Factors

		Value Label	N
gender	1	male	16
	2	female	24
age	1	15-27	22
	2	28-40	18

Table 8.3: Gender and age of the 40 users.

Tests of Within-Subjects Effects

Measure: MEASURE_1

Source		Type III Sum of Squares	df	Mean Square	F	Sig.
sc_pu	Sphericity Assumed	43560927,0	3	14520309,01	119,570	,000
	Greenhouse-Geisser	43560927,0	2,398	18166942,08	119,570	,000
	Huynh-Feldt	43560927,0	2,795	15586234,09	119,570	,000
	Lower-bound	43560927,0	1,000	43560927,03	119,570	,000
sc_pu * gender	Sphericity Assumed	94825,379	3	31608,460	,260	,854
	Greenhouse-Geisser	94825,379	2,398	39546,614	,260	,810
	Huynh-Feldt	94825,379	2,795	33928,813	,260	,840
	Lower-bound	94825,379	1,000	94825,379	,260	,613
sc_pu * age	Sphericity Assumed	559923,101	3	186641,034	1,537	,209
	Greenhouse-Geisser	559923,101	2,398	233514,097	1,537	,217
	Huynh-Feldt	559923,101	2,795	200342,213	1,537	,212
	Lower-bound	559923,101	1,000	559923,101	1,537	,223
sc_pu * gender * age	Sphericity Assumed	692230,585	3	230743,528	1,900	,134
	Greenhouse-Geisser	692230,585	2,398	288692,500	1,900	,148
	Huynh-Feldt	692230,585	2,795	247682,239	1,900	,138
	Lower-bound	692230,585	1,000	692230,585	1,900	,177
Error(sc_pu)	Sphericity Assumed	13115316,6	108	121438,117		
	Greenhouse-Geisser	13115316,6	86,321	151936,107		
	Huynh-Feldt	13115316,6	100,614	130352,798		
	Lower-bound	13115316,6	36,000	364314,350		

Table 8.4: Response_time_mbl, mbl_appl_A (simple), ANOVA table, Tests of Within-Subjects Effects.

Tests of Between-Subjects Effects

Measure: MEASURE_1

Transformed Variable: Average

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Intercept	1391353944	1	1391353944	764,491	,000
gender	1171226,144	1	1171226,144	,644	,428
age	3710469,350	1	3710469,350	2,039	,162
gender * age	7494,693	1	7494,693	,004	,949
Error	65519056,7	36	1819973,797		

Table 8.5: Response_time_mbl, mbl_appl_A (simple), ANOVA table, Tests of Between-Subjects Effects.

From Table 8.4 we take the following results:

- Influence of the factor sc_pu (= combination of scan and push time)
 $F(3,108) = 119,57$ $p = 0,000 < 0,001 < 0,05$ \Rightarrow statistically significant
- Influence of the mutual influence of the factors sc_pu and gender
 $F(3,108) = 0,26$ $p = 0,854 > 0,05$ \Rightarrow statistically insignificant
- Influence of the mutual influence of the factors sc_pu and age
 $F(3,108) = 1,537$ $p = 0,209 > 0,05$ \Rightarrow statistically insignificant
- Influence of the mutual influence of the factors sc_pu, gender and age
 $F(3,108) = 1,900$ $p = 0,134 > 0,05$ \Rightarrow statistically insignificant

From Table 8.5 we take the following results:

- Influence of the factor gender
 $F(1,36) = 0,644$ $p = 0,428 > 0,05$ \Rightarrow statistically insignificant
- Influence of the factor age
 $F(1,36) = 2,039$ $p = 0,162 > 0,05$ \Rightarrow statistically insignificant

We conclude that the combination of scan and push time affects very much the response time of each user. The gender and the age of each user do not affect his or her response time.

8.2.1.3 Gender and Age Graphs

gender	age		SCAN: 5-15sec PUSH: 4-12sec - Resp_time_ mbl_appl_A (in msec)	SCAN: 5-15sec PUSH: 15-23sec - Resp_time_ mbl_appl_A (in msec)	SCAN: 20-30sec PUSH: 4-12sec - Resp_time_ mbl_appl_A (in msec)	SCAN: 20-30sec PUSH: 15-23sec - Resp_time_ mbl_appl_A (in msec)
male	15-27	Mean	2435,00	2924,43	2654,00	3992,43
		N	7	7	7	7
		Std. Deviation	621,591	787,629	540,996	1041,697
		Minimum	1702	1959	2110	2619
		Maximum	3562	4490	3604	5164
		Range	1860	2531	1494	2545
		Median	2191,00	2865,00	2417,00	4269,00
	28-40	Mean	2647,78	3527,44	3072,89	3966,56
		N	9	9	9	9
		Std. Deviation	712,511	872,414	713,917	994,879
		Minimum	1822	2606	1828	2773
		Maximum	3967	5115	4117	5586
		Range	2145	2509	2289	2813
		Median	2420,00	3259,00	3129,00	3803,00
	Total	Mean	2554,69	3263,63	2889,63	3977,87
		N	16	16	16	16
		Std. Deviation	661,206	865,748	659,514	980,874
		Minimum	1702	1959	1828	2619
		Maximum	3967	5115	4117	5586
		Range	2265	3156	2289	2967
		Median	2369,00	2931,00	2689,50	3925,00

female	15-27	Mean	2190,53	2909,13	2583,53	3554,67
		N	15	15	15	15
		Std. Deviation	464,624	704,824	679,423	771,286
		Minimum	1487	1999	1453	2120
		Maximum	2951	4027	4024	5162
		Range	1464	2028	2571	3042
		Median	2228,00	2936,00	2579,00	3452,00
	28-40	Mean	2409,22	3338,89	2802,00	4010,33
		N	9	9	9	9
		Std. Deviation	605,304	789,297	586,417	918,352
		Minimum	1649	2180	1980	2647
		Maximum	3218	4581	3606	5248
		Range	1569	2401	1626	2601
		Median	2245,00	3401,00	2700,00	4221,00
	Total	Mean	2272,54	3070,29	2665,46	3725,54
		N	24	24	24	24
		Std. Deviation	520,134	751,163	642,081	840,374
		Minimum	1487	1999	1453	2120
		Maximum	3218	4581	4024	5248
		Range	1731	2582	2571	3128
		Median	2236,50	3059,00	2692,50	3648,00
Total	15-27	Mean	2268,32	2914,00	2605,95	3693,95
		N	22	22	22	22
		Std. Deviation	517,583	713,080	626,494	866,129
		Minimum	1487	1959	1453	2120
		Maximum	3562	4490	4024	5164
		Range	2075	2531	2571	3044

		Median	2209,50	2900,50	2554,00	3537,50
	28-40	Mean	2528,50	3433,17	2937,44	3988,44
		N	18	18	18	18
		Std. Deviation	652,984	812,866	648,923	929,069
		Minimum	1649	2180	1828	2647
		Maximum	3967	5115	4117	5586
		Range	2318	2935	2289	2939
		Median	2369,00	3330,00	2914,50	3925,00
	Total	Mean	2385,40	3147,63	2755,13	3826,48
		N	40	40	40	40
		Std. Deviation	589,318	793,875	650,226	895,662
		Minimum	1487	1959	1453	2120
		Maximum	3967	5115	4117	5586
		Range	2480	3156	2664	3466
		Median	2281,50	2957,50	2692,50	3723,00

Table 8.6: Total results of Response_time_mbl, mbl_appl_A (simple).

GENDER

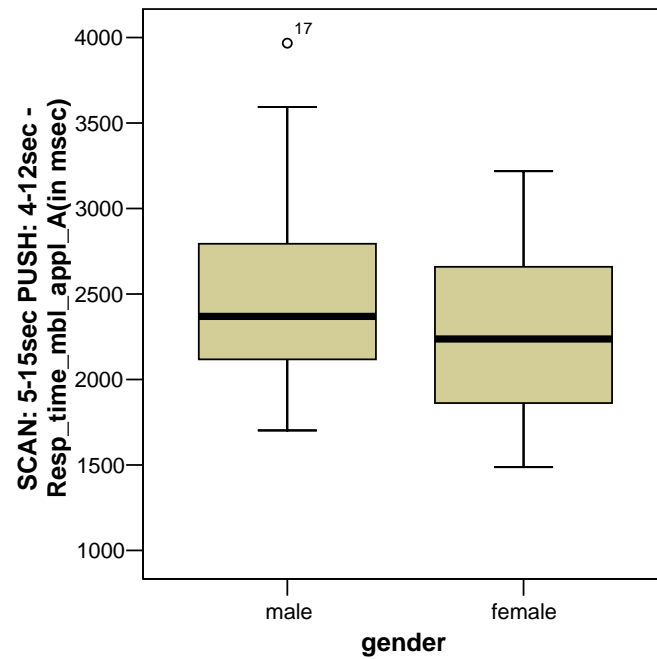


Figure 8.3 : Average Response_time_mbl, mbl_appl_A~gender (shortSCAN-shortPUSH)

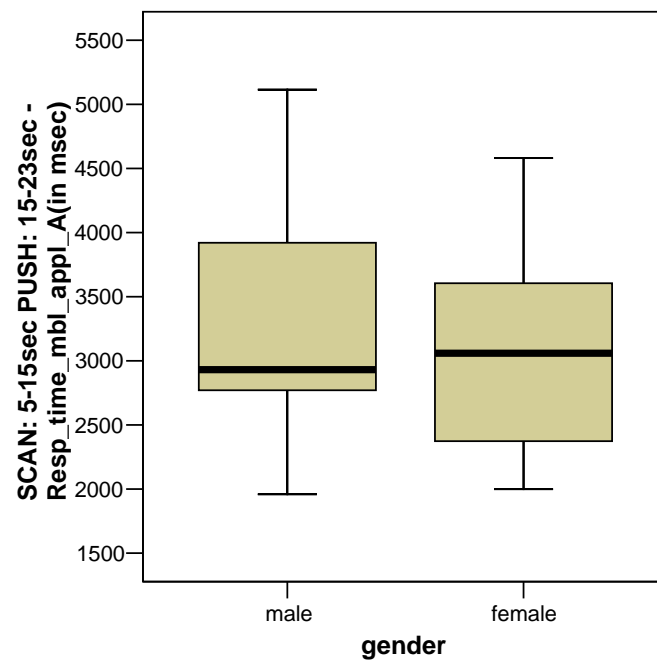


Figure 8.4 : Average Response_time_mbl, mbl_appl_A~gender (shortSCAN-longPUSH)

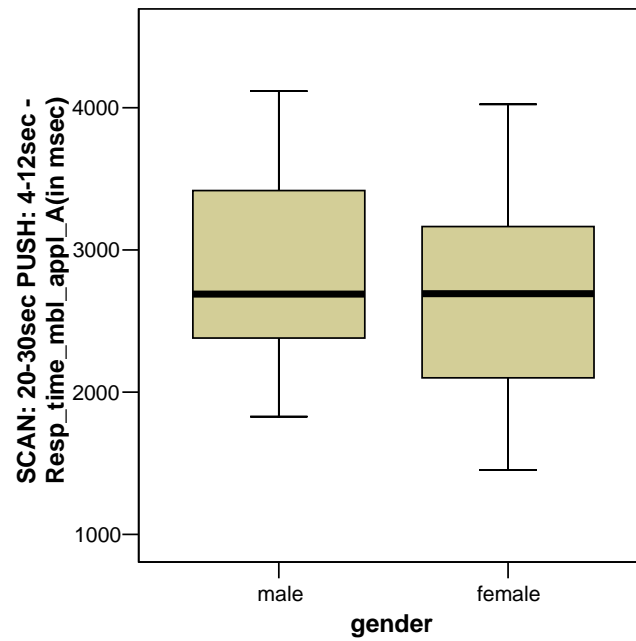


Figure 8.5 : Average Response_time_mbl, mbl_appl_A~gender (longSCAN-shortPUSH)

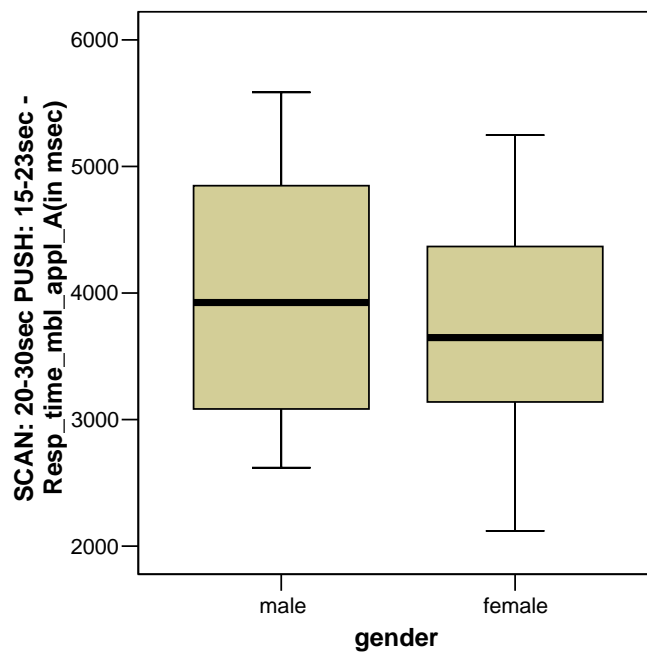


Figure 8.6 : Average Response_time_mbl, mbl_appl_A~gender (longSCAN-longPUSH)

As a result we can see that in the simple application the women response generally a bit faster than the men.

AGE

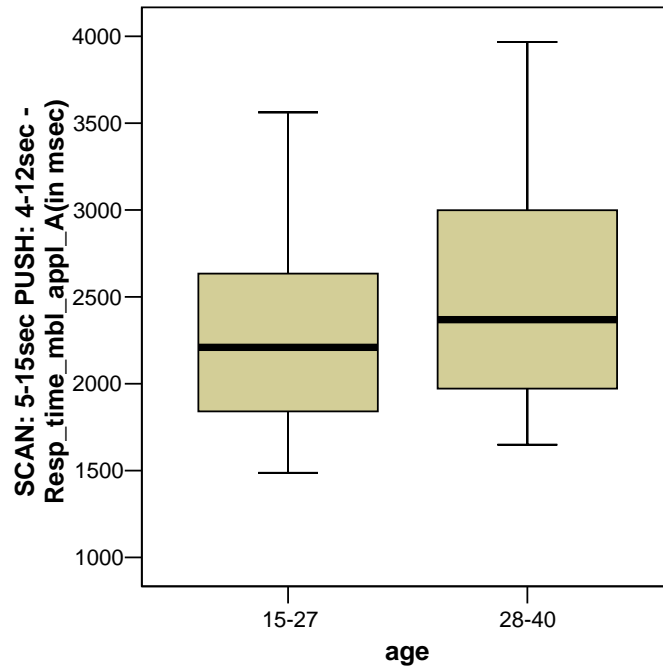


Figure 8.7 : Average Response_time_mbl, mbl_appl_A~age (shortSCAN-shortPUSH)

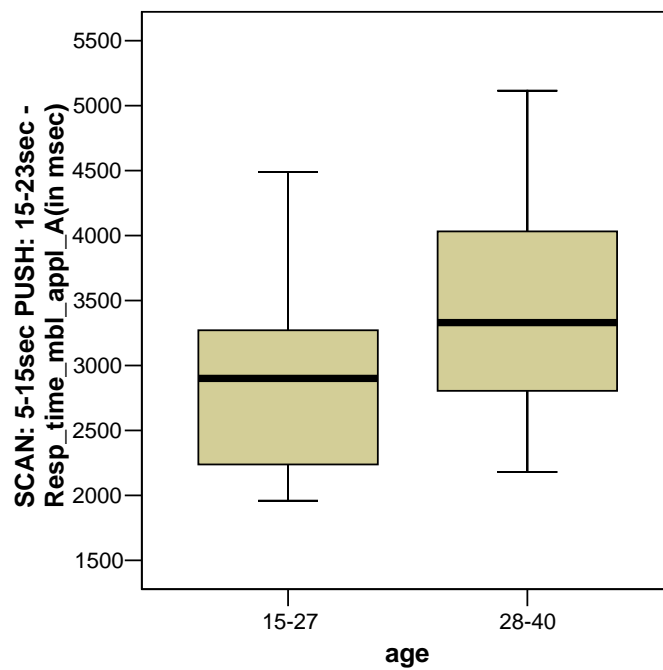


Figure 8.8 : Average Response_time_mbl, mbl_appl_A~age (shortSCAN-longPUSH)

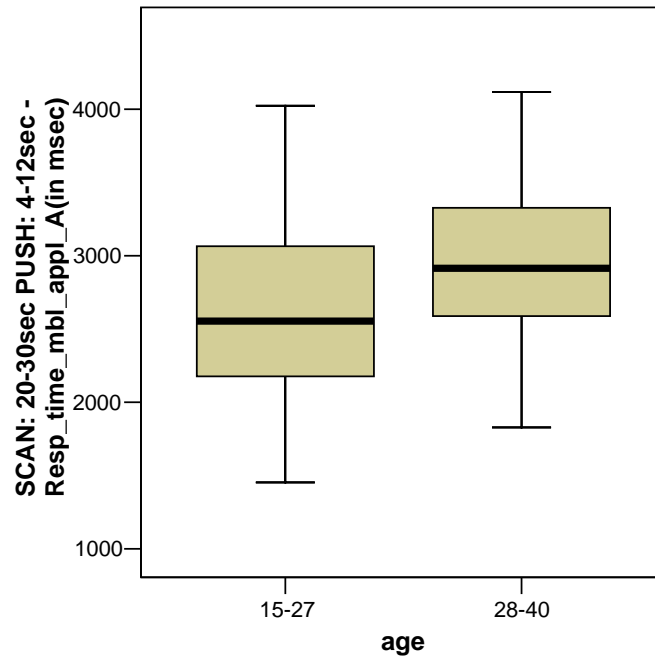


Figure 8.9 : Average Response_time_mbl, mbl_appl_A~age (longSCAN-shortPUSH)

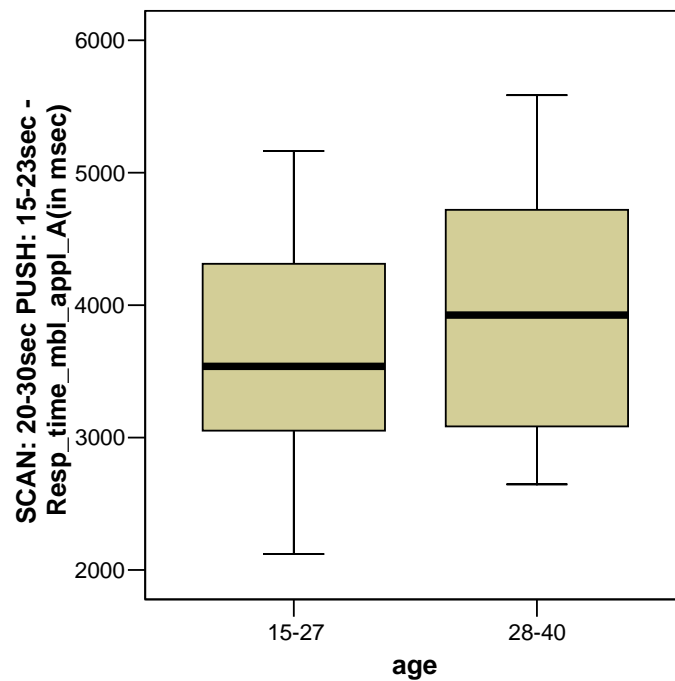


Figure 8.10 : Average Response_time_mbl, mbl_appl_A~age (longSCAN-longPUSH)

As a result the users 15-27 years old response always faster than the users 28-40 years old. This may happen because of the familiarization of younger people with the technology.

8.2.2 Mobile Application B (complex)

8.2.2.1 Scan and Push Times Graphs

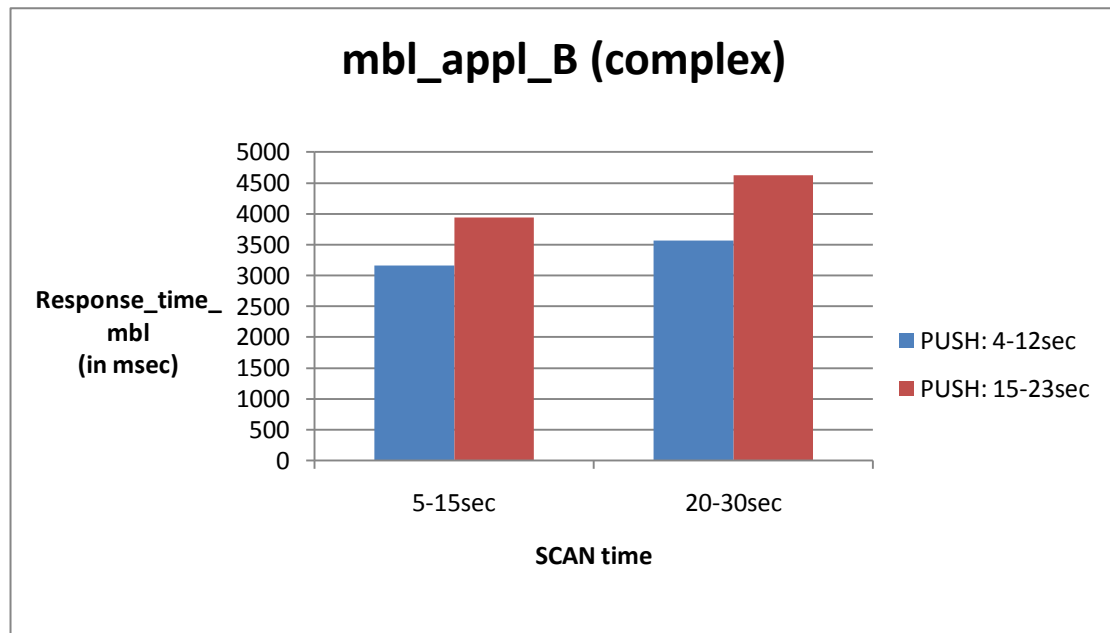


Figure 8.11 : Response_time_mbl, mbl_appl_B (complex)~Scan time .

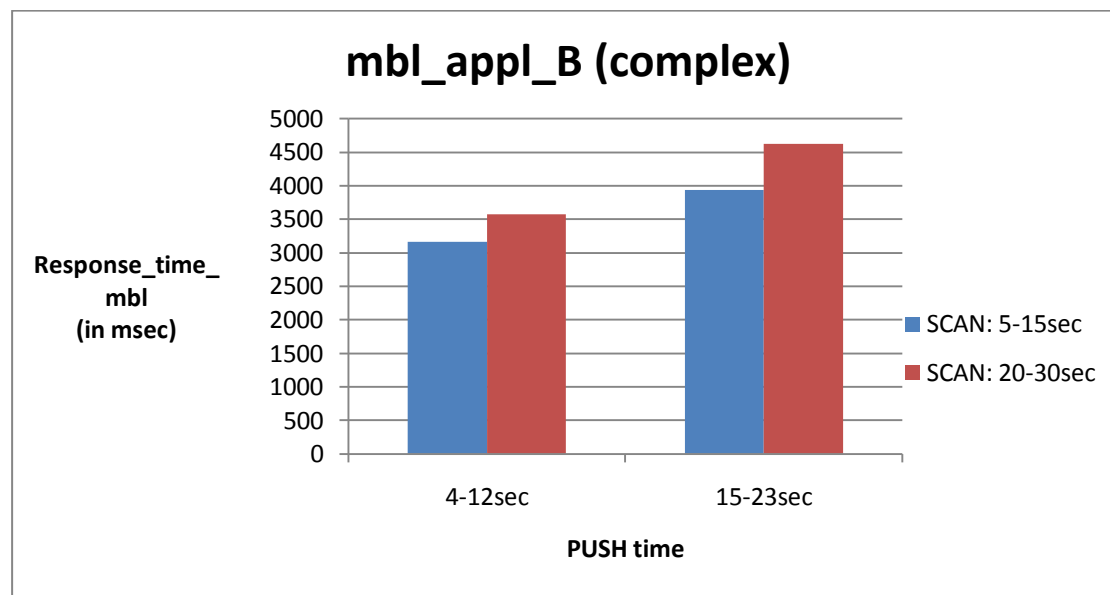


Figure 8.12 : Response_time_mbl, mbl_appl_B (complex)~Push time .

From Figures 8.11, 8.12 we can see that in the second mobile application, which is the complex one, longer scan and push times mean longer response time too.

8.2.2.2 ANOVA

Within-Subjects Factors

Measure: MEASURE_1

sc_pu	Dependent Variable
1	shortSCAN_ shortPUSH
2	shortSCAN_ longPUSH
3	longSCAN_ shortPUSH
4	longSCAN_ longPUSH

Table 8.6 : The four combinations of scan and push times.

Between-Subjects Factors

		Value Label	N
gender	1	male	16
	2	female	24
age	1	15-27	22
	2	28-40	18

Table 8.7 : Gender and age of the 40 users.

Tests of Within-Subjects Effects

Measure: MEASURE_1

Source		Type III Sum of Squares	df	Mean Square	F	Sig.
sc_pu	Sphericity Assumed	46839283,3	3	15613094,42	98,857	,000
	Greenhouse-Geisser	46839283,3	2,701	17341833,50	98,857	,000
	Huynh-Feldt	46839283,3	3,000	15613094,42	98,857	,000
	Lower-bound	46839283,3	1,000	46839283,26	98,857	,000
sc_pu * gender	Sphericity Assumed	781174,751	3	260391,584	1,649	,182
	Greenhouse-Geisser	781174,751	2,701	289223,095	1,649	,188
	Huynh-Feldt	781174,751	3,000	260391,584	1,649	,182
	Lower-bound	781174,751	1,000	781174,751	1,649	,207
sc_pu * age	Sphericity Assumed	692858,133	3	230952,711	1,462	,229
	Greenhouse-Geisser	692858,133	2,701	256524,642	1,462	,232
	Huynh-Feldt	692858,133	3,000	230952,711	1,462	,229
	Lower-bound	692858,133	1,000	692858,133	1,462	,234
sc_pu * gender * age	Sphericity Assumed	538096,474	3	179365,491	1,136	,338
	Greenhouse-Geisser	538096,474	2,701	199225,496	1,136	,336
	Huynh-Feldt	538096,474	3,000	179365,491	1,136	,338
	Lower-bound	538096,474	1,000	538096,474	1,136	,294
Error(sc_pu)	Sphericity Assumed	17057022,7	108	157935,395		
	Greenhouse-Geisser	17057022,7	97,234	175422,582		
	Huynh-Feldt	17057022,7	108,000	157935,395		
	Lower-bound	17057022,7	36,000	473806,186		

Table 8.8 : Response_time_mbl, mbl_appl_B (complex), ANOVA table, Tests of Within-Subjects Effects.

Tests of Between-Subjects Effects

Measure: MEASURE_1

Transformed Variable: Average

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Intercept	2190816369	1	2190816369	1630,350	,000
gender	57537,906	1	57537,906	,043	,837
age	295945,602	1	295945,602	,220	,642
gender * age	408900,948	1	408900,948	,304	,585
Error	48375749,1	36	1343770,810		

Table 8.9 : Response_time_mbl, mbl_appl_B (complex), ANOVA table, Tests of Between-Subjects Effects.

From Table 8.8 we take the following results:

- Influence of the factor sc_pu (= combination of scan and push time)
 $F(3,108) = 98,857$ $p = 0,000 < 0,001 < 0,05 \implies$ statistically significant
- Influence of the mutual influence of the factors sc_pu and gender
 $F(3,108) = 1,649$ $p = 0,182 > 0,05 \implies$ statistically insignificant
- Influence of the mutual influence of the factors sc_pu and age
 $F(3,108) = 1,462$ $p = 0,229 > 0,05 \implies$ statistically insignificant
- Influence of the mutual influence of the factors sc_pu, gender and age
 $F(3,108) = 1,136$ $p = 0,338 > 0,05 \implies$ statistically insignificant

From Table 8.9 we take the following results:

- Influence of the factor gender
 $F(1,36) = 0,043$ $p = 0,837 > 0,05 \implies$ statistically insignificant
- Influence of the factor age
 $F(1,36) = 0,220$ $p = 0,642 > 0,05 \implies$ statistically insignificant

We conclude that the combination of scan and push time affects very much the response time of each user in the complex application too. The gender and the age of each user do not affect his or her response time.

8.2.2.3 Gender and Age Graphs

gender	age		SCAN: 5-15sec PUSH: 4-12sec - Resp_time_ mbl_appl_B (in msec)	SCAN: 5-15sec PUSH: 15-23sec - Resp_time_ mbl_appl_B (in msec)	SCAN: 20-30sec PUSH: 4-12sec - Resp_time_ mbl_appl_B (in msec)	SCAN: 20-30sec PUSH: 15-23sec - Resp_time_ mbl_appl_B (in msec)
male	15-27	Mean	3167,86	3786,57	3709,57	4823,71
		N	7	7	7	7
		Std. Deviation	534,983	543,733	686,979	539,647
		Minimum	2114	3036	2894	4245
		Maximum	3905	4531	4649	5830
		Range	1791	1495	1755	1585
		Median	3251,00	3704,00	3494,00	4762,00
	28-40	Mean	3117,22	3942,78	3554,22	4810,78
		N	9	9	9	9
		Std. Deviation	780,645	817,983	876,145	885,670
		Minimum	2177	2730	2668	3522
		Maximum	4282	5137	4961	6789
		Range	2105	2407	2293	3267
		Median	3001,00	4037,00	3331,00	4541,00
	Total	Mean	3139,38	3874,44	3622,19	4816,44
		N	16	16	16	16
		Std. Deviation	663,455	693,912	777,505	731,358
		Minimum	2114	2730	2668	3522
		Maximum	4282	5137	4961	6789
		Range	2168	2407	2293	3267
		Median	3203,00	4024,00	3429,50	4651,50

female	15-27	Mean	3244,00	3893,93	3453,60	4318,40
		N	15	15	15	15
		Std. Deviation	614,577	721,044	585,391	613,784
		Minimum	2578	2689	2636	3089
		Maximum	4272	5839	4413	5385
		Range	1694	3150	1777	2296
		Median	3058,00	3850,00	3330,00	4434,00
	28-40	Mean	3084,33	4125,11	3668,11	4810,00
		N	9	9	9	9
		Std. Deviation	471,942	615,811	537,348	786,963
		Minimum	2428	3332	2739	3844
		Maximum	3752	5047	4547	5848
		Range	1324	1715	1808	2004
		Median	3166,00	4137,00	3807,00	5074,00
	Total	Mean	3184,13	3980,63	3534,04	4502,75
		N	24	24	24	24
		Std. Deviation	560,012	679,292	565,929	709,810
		Minimum	2428	2689	2636	3089
		Maximum	4272	5839	4547	5848
		Range	1844	3150	1911	2759
		Median	3089,50	3892,50	3444,00	4441,50
Total	15-27	Mean	3219,77	3859,77	3535,05	4479,18
		N	22	22	22	22
		Std. Deviation	578,701	658,553	614,968	626,412
		Minimum	2114	2689	2636	3089
		Maximum	4272	5839	4649	5830
		Range	2158	3150	2013	2741

		Median	3134,00	3795,50	3365,00	4472,00
	28-40	Mean	3100,78	4033,94	3611,17	4810,39
		N	18	18	18	18
		Std. Deviation	626,003	708,609	707,496	812,758
		Minimum	2177	2730	2668	3522
		Maximum	4282	5137	4961	6789
		Range	2105	2407	2293	3267
		Median	3083,50	4087,00	3514,50	4676,00
	Total	Mean	3166,23	3938,15	3569,30	4628,23
		N	40	40	40	40
		Std. Deviation	595,602	678,309	650,614	726,001
		Minimum	2114	2689	2636	3089
		Maximum	4282	5839	4961	6789
		Range	2168	3150	2325	3700
		Median	3134,00	3965,00	3429,50	4533,00

Table 8.10 : Total results of Response_time_mbl, mbl_appl_B (complex).

GENDER

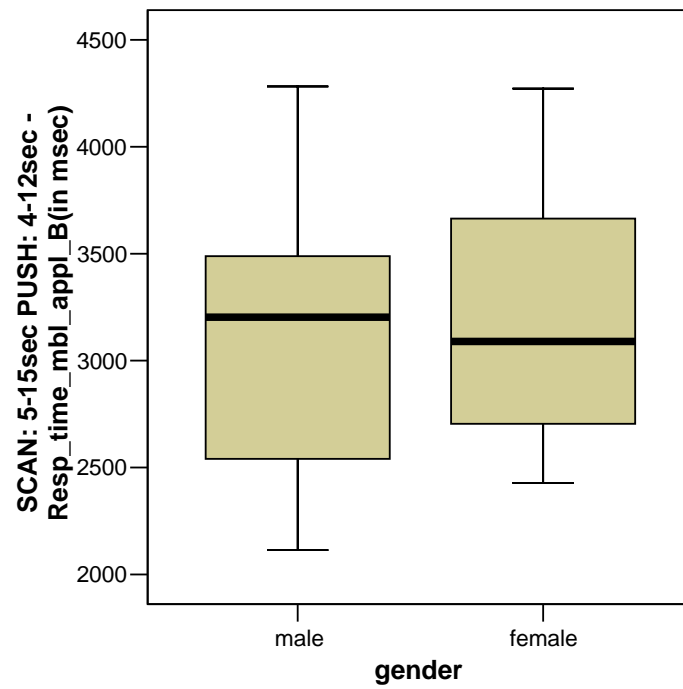


Figure 8.13 : Average Response_time_mbl, mbl_appl_B~gender (shortSCAN-shortPUSH)

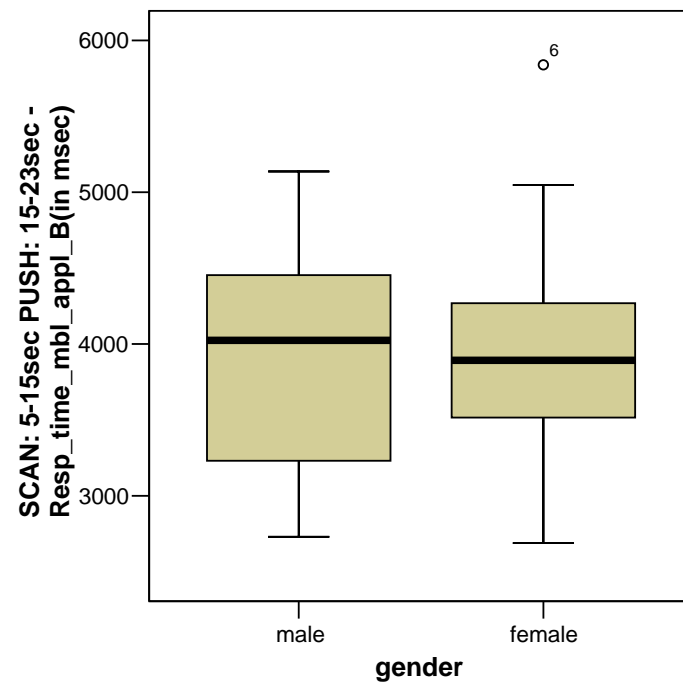


Figure 8.14 : Average Response_time_mbl, mbl_appl_B~gender (shortSCAN-longPUSH)

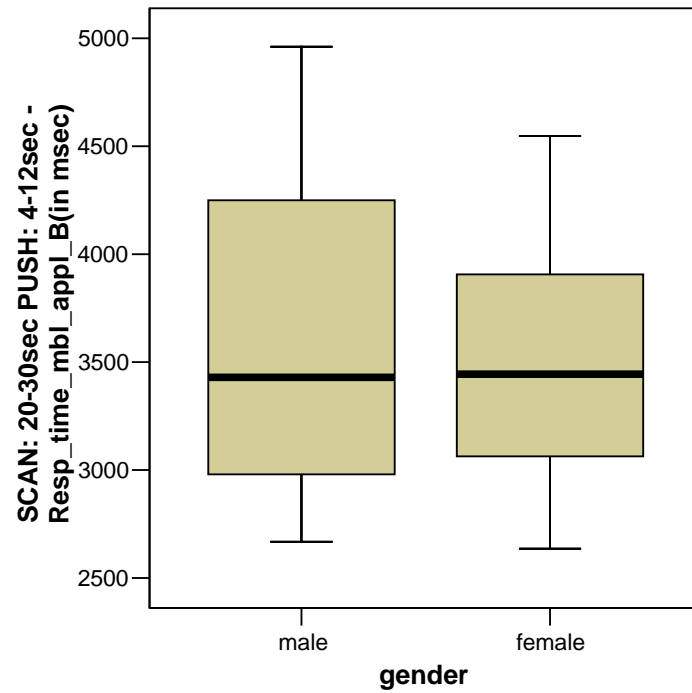


Figure 8.15 : Average Response_time_mbl, mbl_appl_B~gender (longSCAN-shortPUSH)

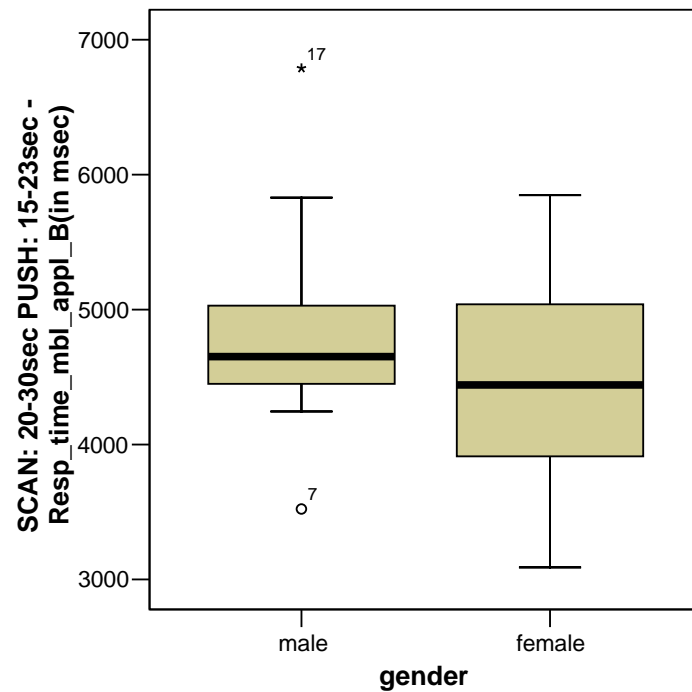


Figure 8.16 : Average Response_time_mbl, mbl_appl_B~gender (longSCAN-longPUSH)

In the complex application the women response generally a bit faster than the men, the same that happens in the simple application.

AGE

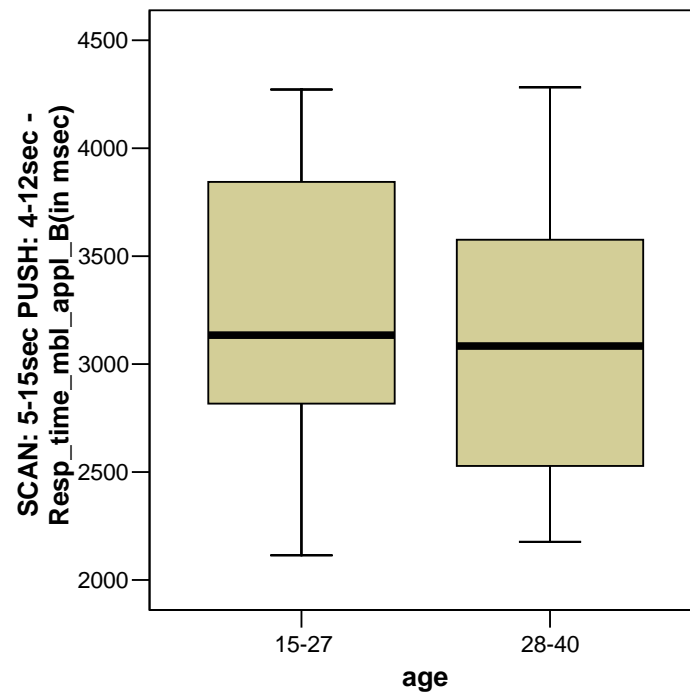


Figure 8.17 : Average Response_time_mbl, mbl_appl_B~age (shortSCAN-shortPUSH)

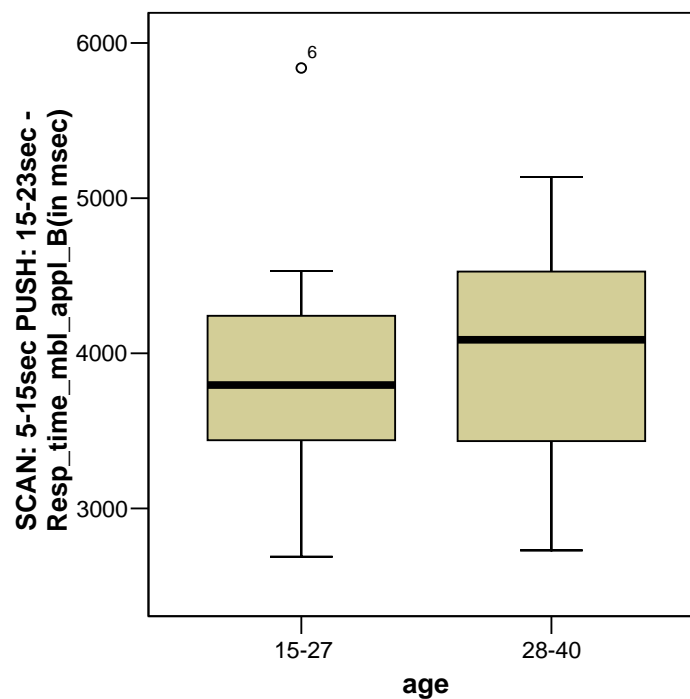


Figure 8.18 : Average Response_time_mbl, mbl_appl_B~age (shortSCAN-longPUSH)

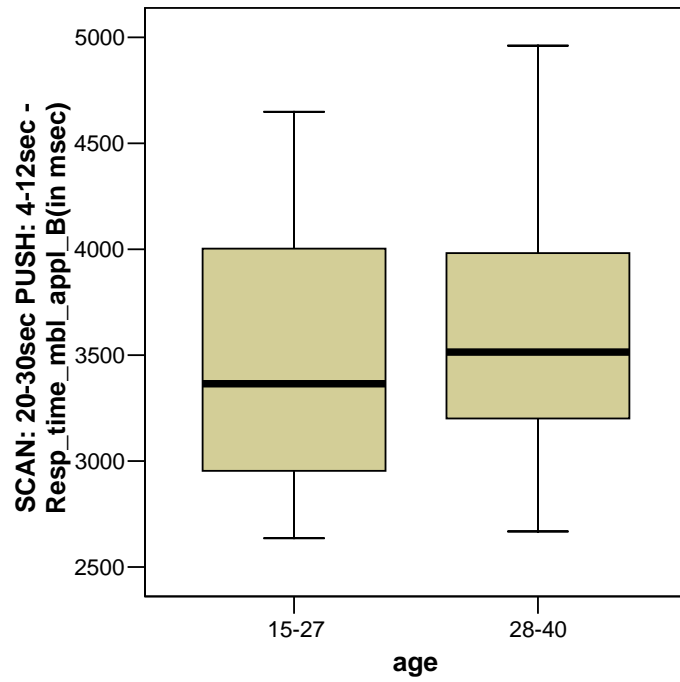


Figure 8.19 : Average Response_time_mbl, mbl_appl_B~age (longSCAN-shortPUSH)

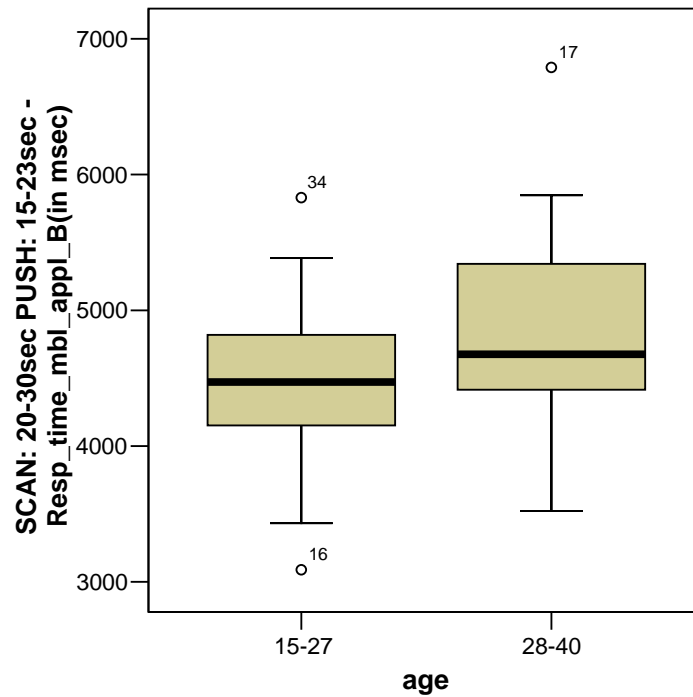


Figure 8.20 : Average Response_time_mbl, mbl_appl_B~age (longSCAN-longPUSH)

As a result the users 15-27 years old response always faster than the users 28-40 years old.

8.3 Penalty Points Mobile

8.3.1 Mobile Application A (simple)

8.3.1.1 Scan and Push Times Graphs

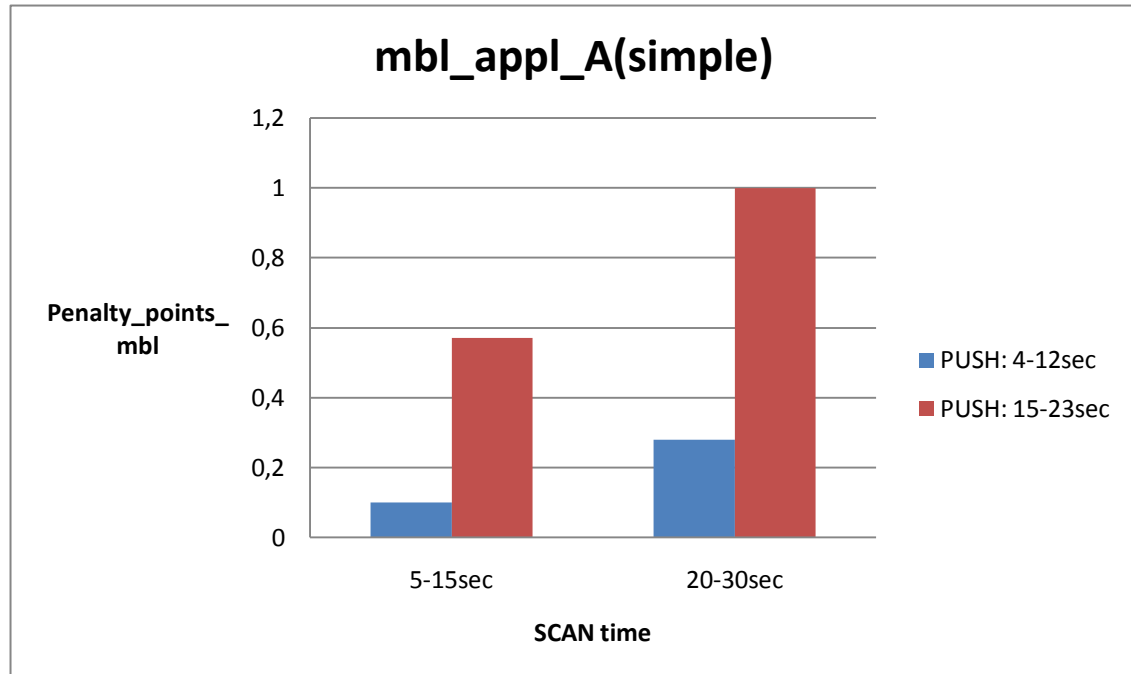


Figure 8.21 : Penalty_points_mbl, mbl_appl_A (simple)~Scan time .

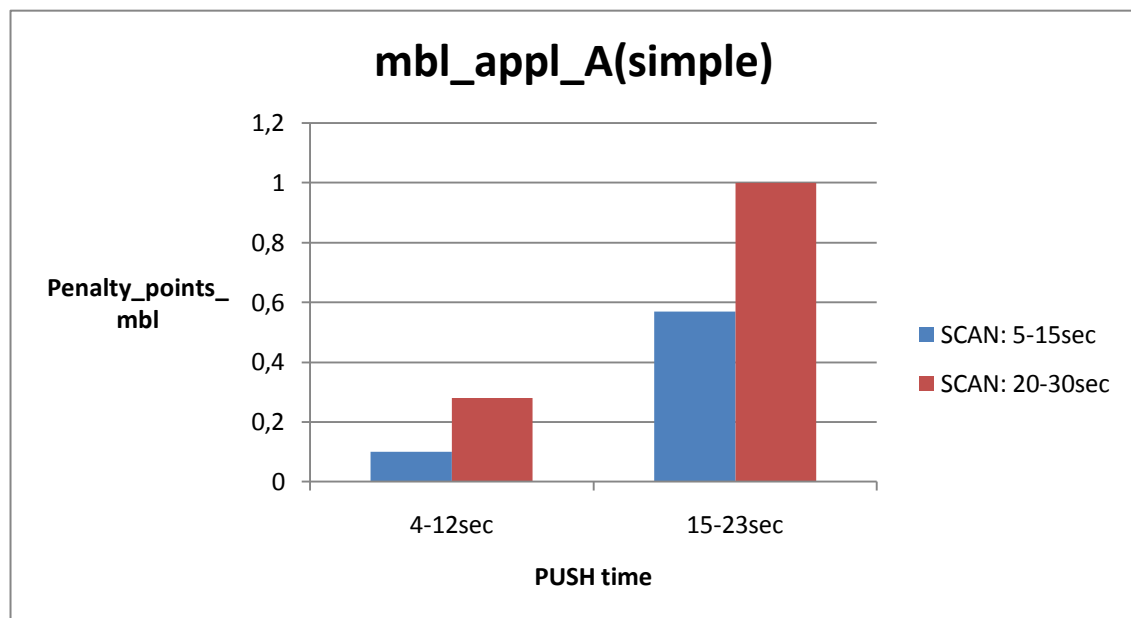


Figure 8.22 : Penalty_points_mbl, mbl_appl_A (simple)~Push time .

From Figures 8.21, 8.22 we conclude that in the simple application, longer scan and push times mean more mistakes during the execution of the instructions the user receives to his phone.

8.3.1.2 ANOVA

Within-Subjects Factors

Measure: MEASURE_1

sc_pu	Dependent Variable
1	shortSCAN_ shortPUSH
2	shortSCSN_ longPUSH
3	longSCSN_ shortPUSH
4	longSCAN_ longPUSH

Table 8.11 : The four combinations of scan and push times.

Between-Subjects Factors

	Value Label	N
gender 1	male	16
2	female	24
age 1	15-27	22
2	28-40	18

Table 8.12 : Gender and age of the 40 users.

Tests of Within-Subjects Effects

Measure: MEASURE_1

Source		Type III Sum of Squares	df	Mean Square	F	Sig.
sc_pu	Sphericity Assumed	15,636	3	5,212	13,512	,000
	Greenhouse-Geisser	15,636	2,358	6,632	13,512	,000
	Huynh-Feldt	15,636	2,744	5,699	13,512	,000
	Lower-bound	15,636	1,000	15,636	13,512	,001
sc_pu * gender	Sphericity Assumed	1,574	3	,525	1,360	,259
	Greenhouse-Geisser	1,574	2,358	,668	1,360	,262
	Huynh-Feldt	1,574	2,744	,574	1,360	,261
	Lower-bound	1,574	1,000	1,574	1,360	,251
sc_pu * age	Sphericity Assumed	,754	3	,251	,651	,584
	Greenhouse-Geisser	,754	2,358	,320	,651	,548
	Huynh-Feldt	,754	2,744	,275	,651	,571
	Lower-bound	,754	1,000	,754	,651	,425
sc_pu * gender * age	Sphericity Assumed	,378	3	,126	,327	,806
	Greenhouse-Geisser	,378	2,358	,160	,327	,757
	Huynh-Feldt	,378	2,744	,138	,327	,788
	Lower-bound	,378	1,000	,378	,327	,571
Error(sc_pu)	Sphericity Assumed	41,660	108	,386		
	Greenhouse-Geisser	41,660	84,872	,491		
	Huynh-Feldt	41,660	98,771	,422		
	Lower-bound	41,660	36,000	1,157		

Table 8.13 : Penalty_points_mbl, mbl_appl_A (simple), ANOVA table, Tests of Within-Subjects Effects.

Tests of Between-Subjects Effects

Measure: MEASURE_1

Transformed Variable: Average

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Intercept	35,950	1	35,950	33,391	,000
gender	,002	1	,002	,002	,967
age	,185	1	,185	,172	,681
gender * age	,002	1	,002	,002	,967
Error	38,759	36	1,077		

Table 8.14 : Penalty_points_mbl, mbl_appl_A (simple), ANOVA table, Tests of Between-Subjects Effects.

From Table 8.13 we take the following results:

- Influence of the factor sc_pu (= combination of scan and push time)

$$F(3,108) = 13,512 \quad p = 0,000 < 0,001 < 0,05 \quad \Rightarrow \text{statistically significant}$$

- Influence of the mutual influence of the factors sc_pu and gender

$$F(3,108) = 1,360 \quad p = 0,259 > 0,05 \quad \Rightarrow \text{statistically insignificant}$$

- Influence of the mutual influence of the factors sc_pu and age

$$F(3,108) = 0,651 \quad p = 0,584 > 0,05 \quad \Rightarrow \text{statistically insignificant}$$

- Influence of the mutual influence of the factors sc_pu, gender and age

$$F(3,108) = 0,327 \quad p = 0,806 > 0,05 \quad \Rightarrow \text{statistically insignificant}$$

From Table 8.14 we take the following results:

- Influence of the factor gender

$$F(1,36) = 0,002 \quad p = 0,967 > 0,05 \quad \Rightarrow \text{statistically insignificant}$$

- Influence of the factor age

$$F(1,36) = 0,172 \quad p = 0,681 > 0,05 \quad \Rightarrow \text{statistically insignificant}$$

We conclude that the combination of scan and push time affects very much the mistakes that each user makes in the simple application as he executes the instructions he receives. The gender and the age of each user do not affect the penalty points.

8.3.1.3 Gender and Age Graphs

gender	age		SCAN: 5-15sec PUSH: 4-12sec Penalty_points_mbl_appl_A	SCAN: 5-15sec PUSH: 15-23sec Penalty_points_mbl_appl_A	SCAN: 20-30sec PUSH: 4-12sec Penalty_points_mbl_appl_A	SCAN: 20-30sec PUSH: 15-23sec Penalty_points_mbl_appl_A
male	15-27	Mean	,00	,43	,57	,86
		N	7	7	7	7
		Std. Deviation	,000	,787	1,134	1,215
		Minimum	0	0	0	0
		Maximum	0	2	3	3
		Range	0	2	3	3
		Median	,00	,00	,00	,00
	28-40	Mean	,22	,78	,33	,78
		N	9	9	9	9
		Std. Deviation	,667	,833	,500	,441
		Minimum	0	0	0	0
		Maximum	2	2	1	1
		Range	2	2	1	1
		Median	,00	1,00	,00	1,00
	Total	Mean	,13	,63	,44	,81
		N	16	16	16	16
		Std. Deviation	,500	,806	,814	,834
		Minimum	0	0	0	0
		Maximum	2	2	3	3
		Range	2	2	3	3
		Median	,00	,00	,00	1,00

female	15-27	Mean	,07	,47	,13	1,13
		N	15	15	15	15
		Std. Deviation	,258	,743	,352	1,060
		Minimum	0	0	0	0
		Maximum	1	2	1	4
		Range	1	2	1	4
		Median	,00	,00	,00	1,00
	28-40	Mean	,11	,67	,22	1,11
		N	9	9	9	9
		Std. Deviation	,333	1,000	,441	1,167
		Minimum	0	0	0	0
		Maximum	1	2	1	3
		Range	1	2	1	3
		Median	,00	,00	,00	1,00
	Total	Mean	,08	,54	,17	1,13
		N	24	24	24	24
		Std. Deviation	,282	,833	,381	1,076
		Minimum	0	0	0	0
		Maximum	1	2	1	4
		Range	1	2	1	4
		Median	,00	,00	,00	1,00
Total	15-27	Mean	,05	,45	,27	1,05
		N	22	22	22	22
		Std. Deviation	,213	,739	,703	1,090
		Minimum	0	0	0	0
		Maximum	1	2	3	4
		Range	1	2	3	4

	28-40	Median	,00	,00	,00	1,00
		Mean	,17	,72	,28	,94
		N	18	18	18	18
		Std. Deviation	,514	,895	,461	,873
		Minimum	0	0	0	0
		Maximum	2	2	1	3
		Range	2	2	1	3
		Median	,00	,00	,00	1,00
	Total	Mean	,10	,57	,28	1,00
		N	40	40	40	40
		Std. Deviation	,379	,813	,599	,987
		Minimum	0	0	0	0
		Maximum	2	2	3	4
		Range	2	2	3	4
		Median	,00	,00	,00	1,00

Table 8.15 : Total results of Penalty_points_mbl, mbl_appl_A (simple).

GENDER

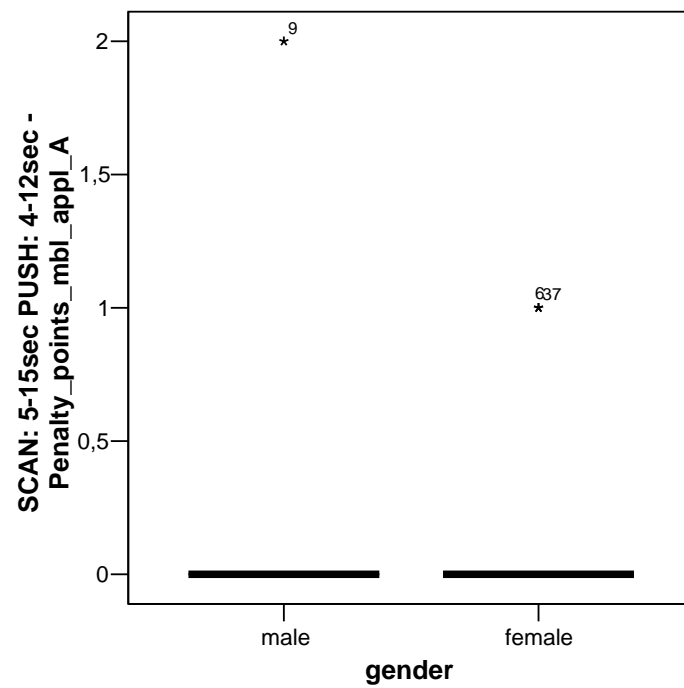


Figure 8.23 : Average Penalty_points_mbl, mbl_appl_A~gender (shortSCAN-shortPUSH)

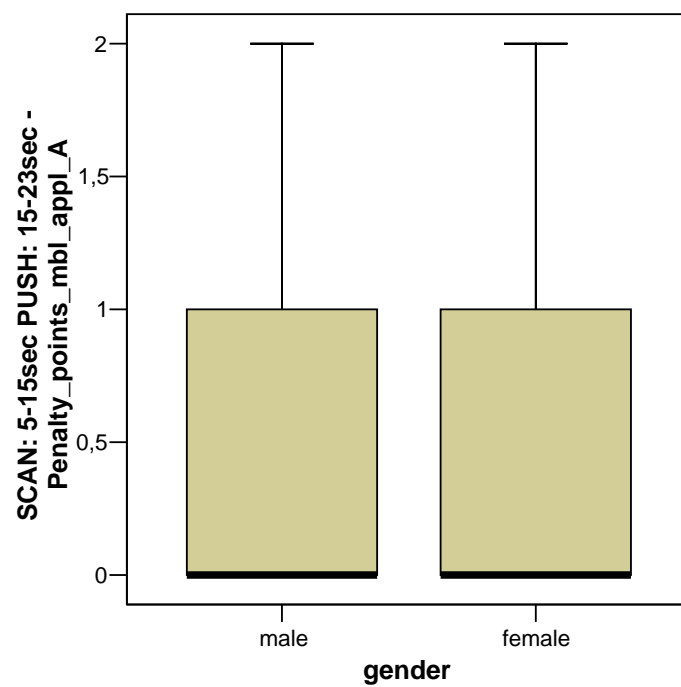


Figure 8.24 : Average Penalty_points_mbl, mbl_appl_A~gender (shortSCAN-longPUSH)

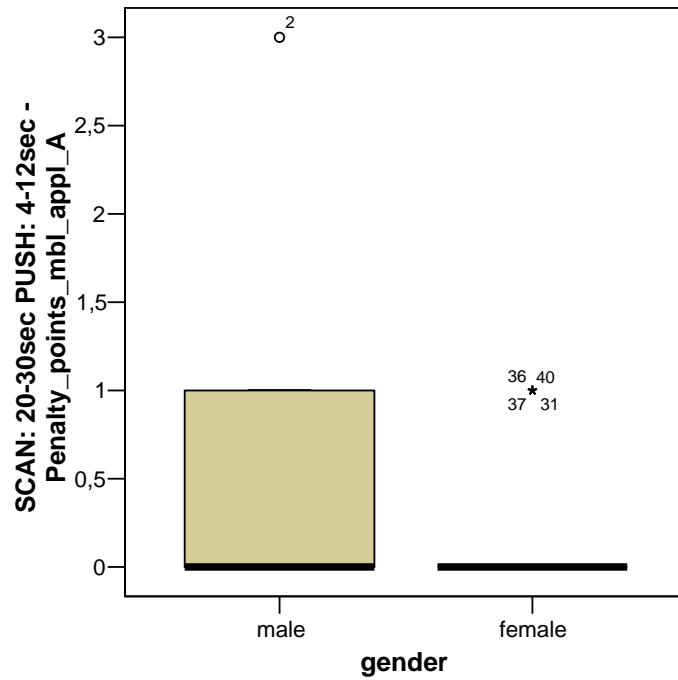


Figure 8.25 : Average Penalty_points_mbl, mbl_appl_A~gender (longSCAN-shortPUSH)

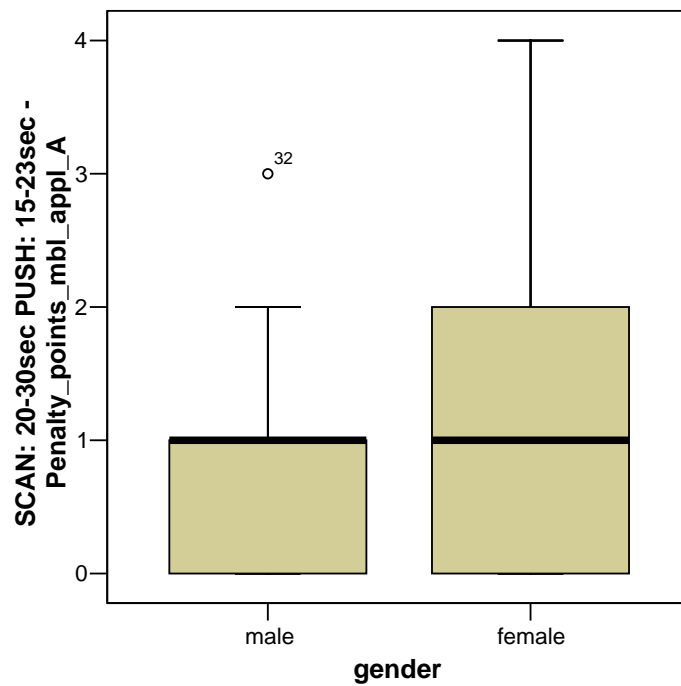


Figure 8.26 : Average Penalty_points_mbl, mbl_appl_A~gender (longSCAN-longPUSH)

In the simple application men and women have the same average penalty points during the execution of the instructions they receive.

AGE

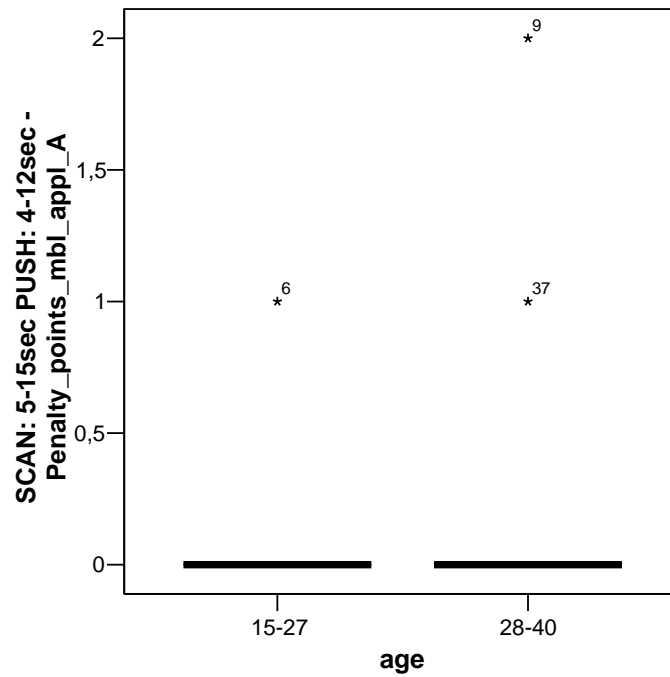


Figure 8.27 : Average Penalty_points_mbl, mbl_appl_A~age
(shortSCAN-shortPUSH)

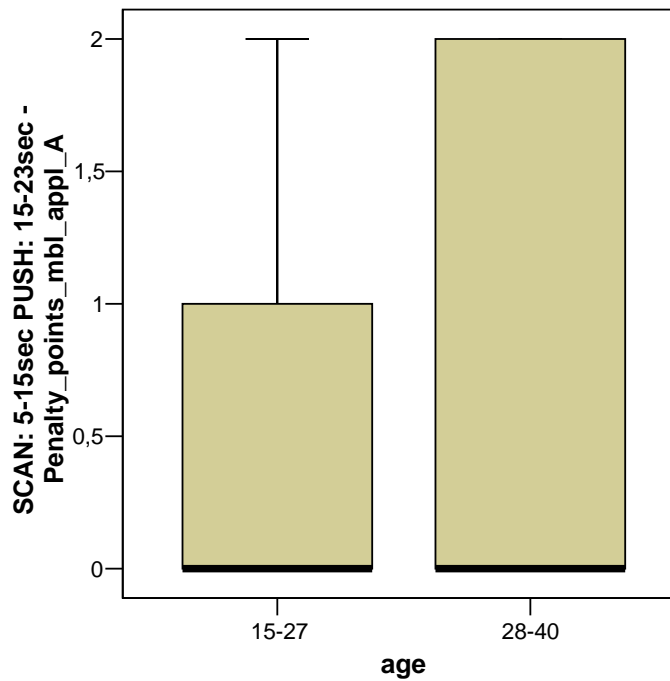


Figure 8.28 : Average Penalty_points_mbl, mbl_appl_A~age
(shortSCAN-longPUSH)

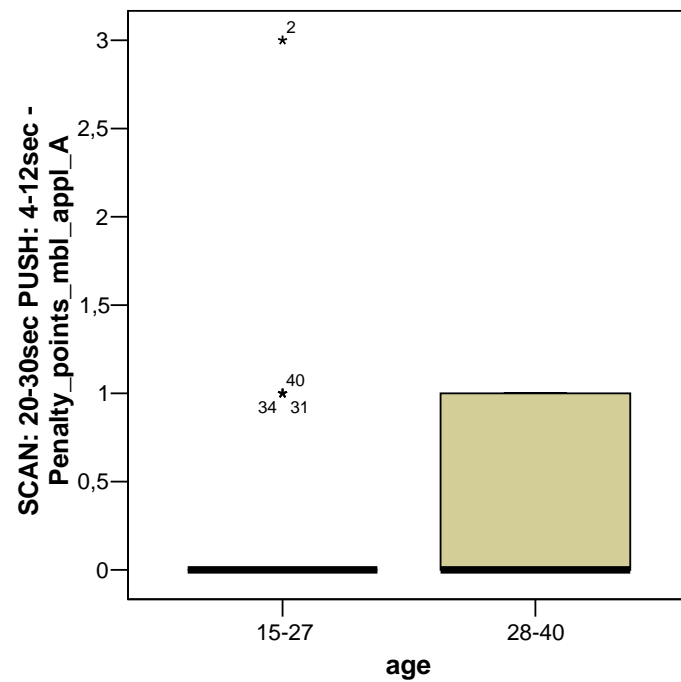


Figure 8.29 : Average Penalty_points_mbl, mbl_appl_A~age (longSCAN-shortPUSH)

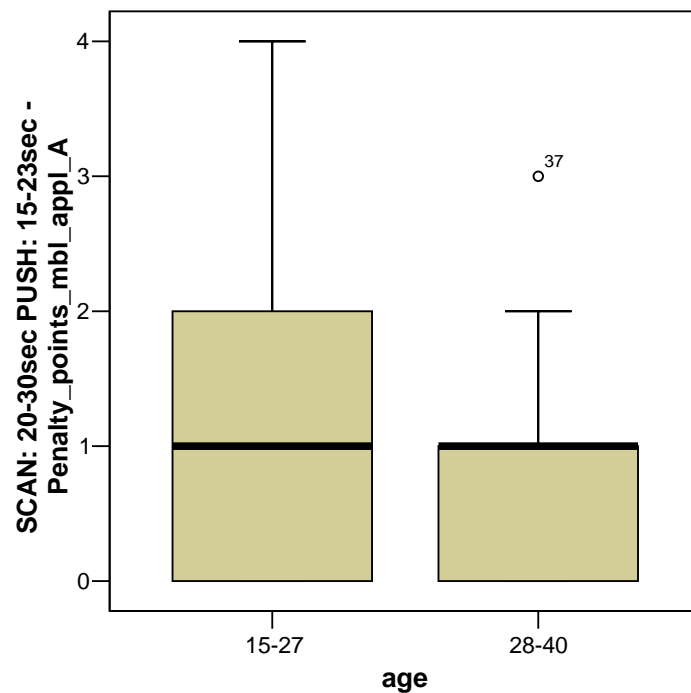


Figure 8.30 : Average Penalty_points_mbl, mbl_appl_A~age (longSCAN-longPUSH)

As a result the users 15-27 years and also the users 28-40 years old have the same average penalty points on the mobile phone in the simple application.

8.3.2 Mobile Application B (complex)

8.3.2.1 Scan and Push Times Graphs

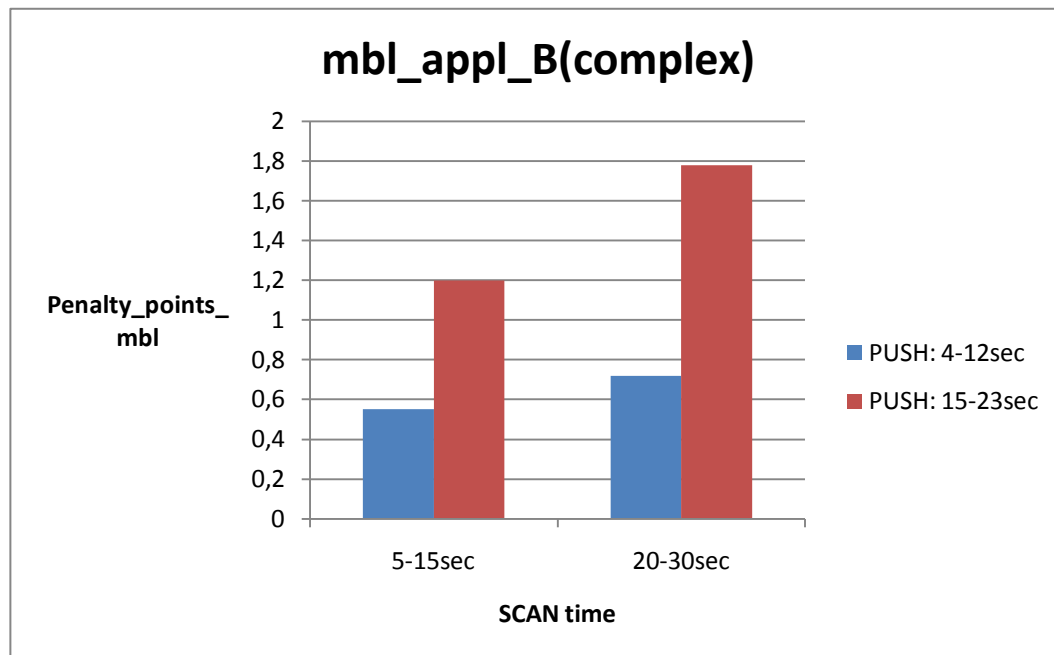


Figure 8.31 : Penalty_points_mbl, mbl_appl_B (complex)~Scan time .

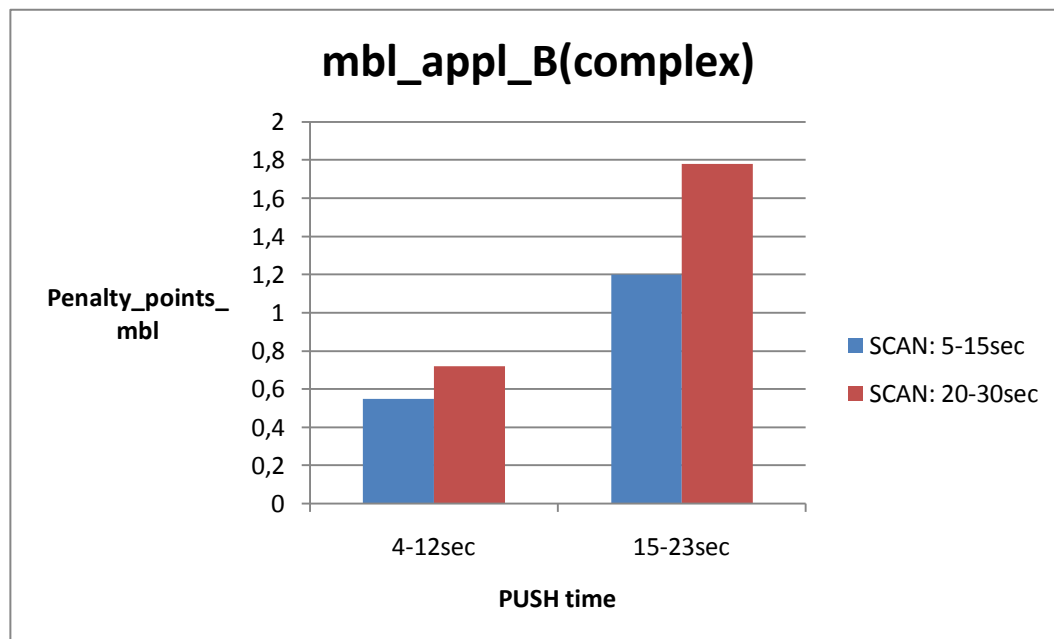


Figure 8.32 : Penalty_points_mbl, mbl_appl_B (complex)~Push time .

From Figures 8.31, 8.32 we conclude that in the complex application, longer scan and push times mean more mistakes during the execution of the instructions the user receives to his phone.

8.3.2.2 ANOVA

Within-Subjects Factors

Measure: MEASURE_1

sc pu	Dependent Variable
1	shortSCAN_ shortPUSH
2	shortSCAN_ longPUSH
3	longSCAN_ shortPUSH
4	longSCAN_ longPUSH

Table 8.16 : The four combinations of scan and push times.

Between-Subjects Factors

	Value Label	N
gender 1	male	16
2	female	24
age 1	15-27	22
2	28-40	18

Table 8.17 : Gender and age of the 40 users.

Tests of Within-Subjects Effects

Measure: MEASURE_1

Source		Type III Sum of Squares	df	Mean Square	F	Sig.
factor1	Sphericity Assumed	19,023	3	6,341	13,506	,000
	Greenhouse-Geisser	19,023	2,195	8,668	13,506	,000
	Huynh-Feldt	19,023	2,538	7,497	13,506	,000
	Lower-bound	19,023	1,000	19,023	13,506	,001
factor1 * gender	Sphericity Assumed	2,202	3	,734	1,563	,202
	Greenhouse-Geisser	2,202	2,195	1,003	1,563	,214
	Huynh-Feldt	2,202	2,538	,868	1,563	,209
	Lower-bound	2,202	1,000	2,202	1,563	,219
factor1 * age	Sphericity Assumed	1,202	3	,401	,853	,468
	Greenhouse-Geisser	1,202	2,195	,548	,853	,439
	Huynh-Feldt	1,202	2,538	,474	,853	,452
	Lower-bound	1,202	1,000	1,202	,853	,362
factor1 * gender * age	Sphericity Assumed	,494	3	,165	,351	,789
	Greenhouse-Geisser	,494	2,195	,225	,351	,725
	Huynh-Feldt	,494	2,538	,195	,351	,755
	Lower-bound	,494	1,000	,494	,351	,557
Error(factor1)	Sphericity Assumed	50,706	108	,470		
	Greenhouse-Geisser	50,706	79,006	,642		
	Huynh-Feldt	50,706	91,353	,555		
	Lower-bound	50,706	36,000	1,409		

Table 8.18 : Penalty_points_mbl, mbl_appl_B (complex), ANOVA table, Tests of Within-Subjects Effects.

Tests of Between-Subjects Effects

Measure: MEASURE_1

Transformed Variable: Average

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Intercept	125,531	1	125,531	60,137	,000
gender	5,245	1	5,245	2,512	,122
age	2,186	1	2,186	1,047	,313
gender * age	1,590	1	1,590	,762	,389
Error	75,148	36	2,087		

Table 8.19 : Penalty_points_mbl, mbl_appl_B (complex), ANOVA table, Tests of Between-Subjects Effects.

From Table 8.18 we take the following results:

- Influence of the factor sc_pu (= combination of scan and push time)
 $F(3,108) = 13,506$ $p = 0,000 < 0,001 < 0,05$ \Rightarrow statistically significant
- Influence of the mutual influence of the factors sc_pu and gender
 $F(3,108) = 1,563$ $p = 0,202 > 0,05$ \Rightarrow statistically insignificant
- Influence of the mutual influence of the factors sc_pu and age
 $F(3,108) = 0,853$ $p = 0,468 > 0,05$ \Rightarrow statistically insignificant
- Influence of the mutual influence of the factors sc_pu, gender and age
 $F(3,108) = 0,351$ $p = 0,789 > 0,05$ \Rightarrow statistically insignificant

From Table 8.19 we take the following results:

- Influence of the factor gender
 $F(1,36) = 2,512$ $p = 0,122 > 0,05$ \Rightarrow statistically insignificant
- Influence of the factor age
 $F(1,36) = 1,047$ $p = 0,313 > 0,05$ \Rightarrow statistically insignificant

We conclude that the combination of scan and push time affects very much the mistakes that each user makes in the complex application as he executes the instructions he receives. The gender and the age of each user do not affect the penalty points.

8.3.2.3 Gender and Age Graphs

gender	age		SCAN: 5-15sec PUSH: 4-12sec Penalty_point s_mbl_appl_ B	SCAN: 5-15sec PUSH: 15-23sec Penalty_point s_mbl_appl_ B	SCAN: 20-30sec PUSH: 4-12sec Penalty_point s_mbl_appl_ B	SCAN: 20-30sec PUSH: 15-23sec Penalty_point s_mbl_appl_ B
male	15-27	Mean	,14	,71	,57	1,29
		N	7	7	7	7
		Std. Deviation	,378	1,113	,787	1,604
		Minimum	0	0	0	0
		Maximum	1	3	2	4
		Range	1	3	2	4
		Median	,00	,00	,00	1,00
	28-40	Mean	,44	1,00	,67	1,11
		N	9	9	9	9
		Std. Deviation	,527	1,118	,866	,928
		Minimum	0	0	0	0
		Maximum	1	3	2	2
		Range	1	3	2	2
		Median	,00	1,00	,00	1,00
	Total	Mean	,31	,88	,63	1,19
		N	16	16	16	16
		Std. Deviation	,479	1,088	,806	1,223
		Minimum	0	0	0	0
		Maximum	1	3	2	4
		Range	1	3	2	4
		Median	,00	,50	,00	1,00

female	15-27	Mean	,40	,93	,47	1,67
		N	15	15	15	15
		Std. Deviation	,507	,884	,516	1,175
		Minimum	0	0	0	0
		Maximum	1	2	1	4
		Range	1	2	1	4
		Median	,00	1,00	,00	1,00
	28-40	Mean	1,22	2,22	1,33	3,00
		N	9	9	9	9
		Std. Deviation	,667	,833	,866	,707
		Minimum	0	1	0	2
		Maximum	2	3	3	4
		Range	2	2	3	2
		Median	1,00	2,00	1,00	3,00
	Total	Mean	,71	1,42	,79	2,17
		N	24	24	24	24
		Std. Deviation	,690	1,060	,779	1,204
		Minimum	0	0	0	0
		Maximum	2	3	3	4
		Range	2	3	3	4
		Median	1,00	1,50	1,00	2,00
Total	15-27	Mean	,32	,86	,50	1,55
		N	22	22	22	22
		Std. Deviation	,477	,941	,598	1,299
		Minimum	0	0	0	0
		Maximum	1	3	2	4
		Range	1	3	2	4

	28-40	Median	,00	1,00	,00	1,00
		Mean	,83	1,61	1,00	2,06
		N	18	18	18	18
		Std. Deviation	,707	1,145	,907	1,259
		Minimum	0	0	0	0
		Maximum	2	3	3	4
		Range	2	3	3	4
		Median	1,00	2,00	1,00	2,00
	Total	Mean	,55	1,20	,72	1,78
		N	40	40	40	40
		Std. Deviation	,639	1,091	,784	1,291
		Minimum	0	0	0	0
		Maximum	2	3	3	4
		Range	2	3	3	4
		Median	,00	1,00	1,00	2,00

Table 8.20 : Total results of Penalty_points_mbl, mbl_appl_B (complex).

GENDER

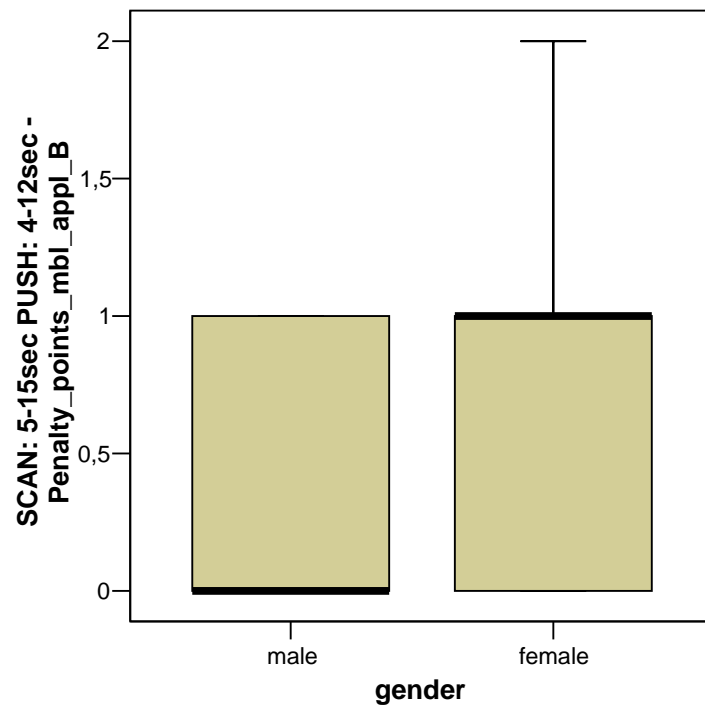


Figure 8.33 : Average Penalty_points_mbl, mbl_appl_B~gender (shortSCAN-shortPUSH)

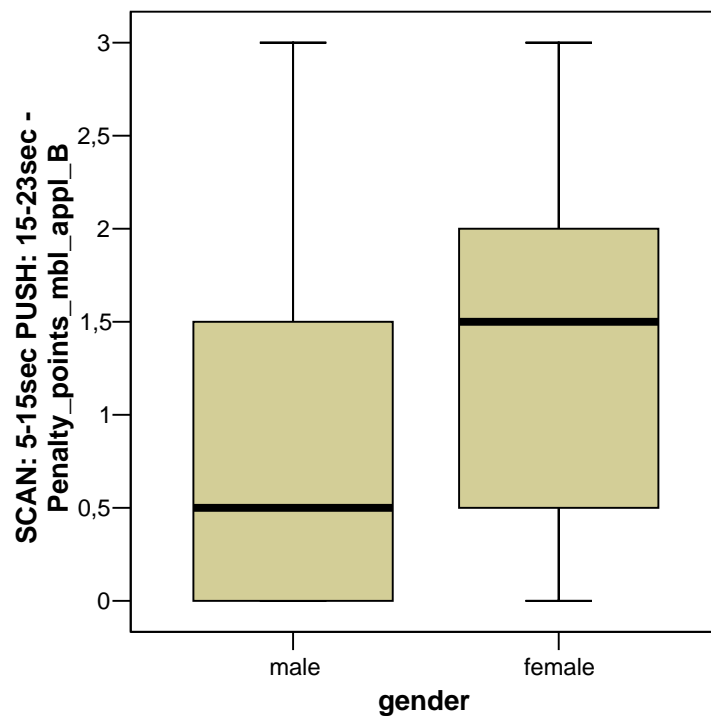


Figure 8.34 : Average Penalty_points_mbl, mbl_appl_B~gender (shortSCAN-longPUSH)

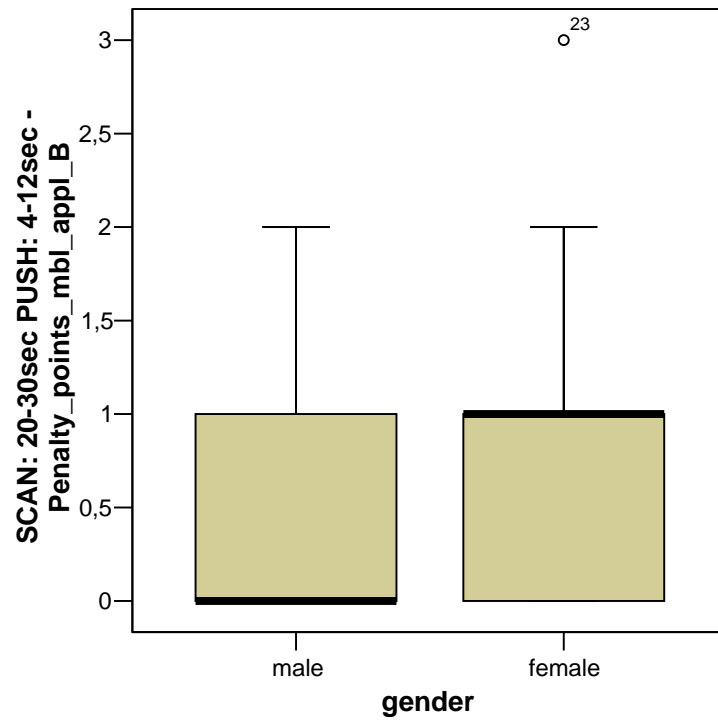


Figure 8.35 : Average Penalty_points_mbl, mbl_appl_B~gender (longSCAN-shortPUSH)

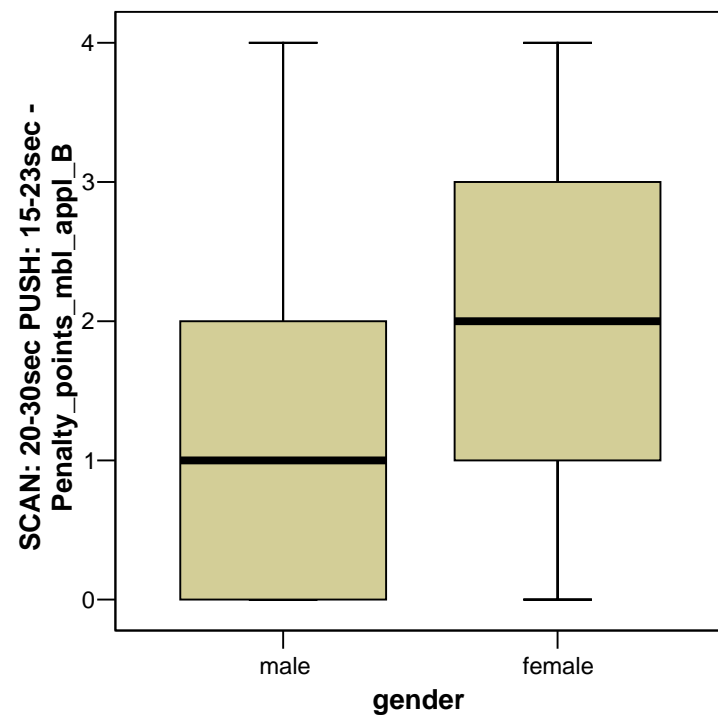


Figure 8.36 : Average Penalty_points_mbl, mbl_appl_B~gender (longSCAN-longPUSH)

In the complex application the women, that response faster as we saw earlier, tend to make more mistakes when the complexity of the application is higher.

AGE

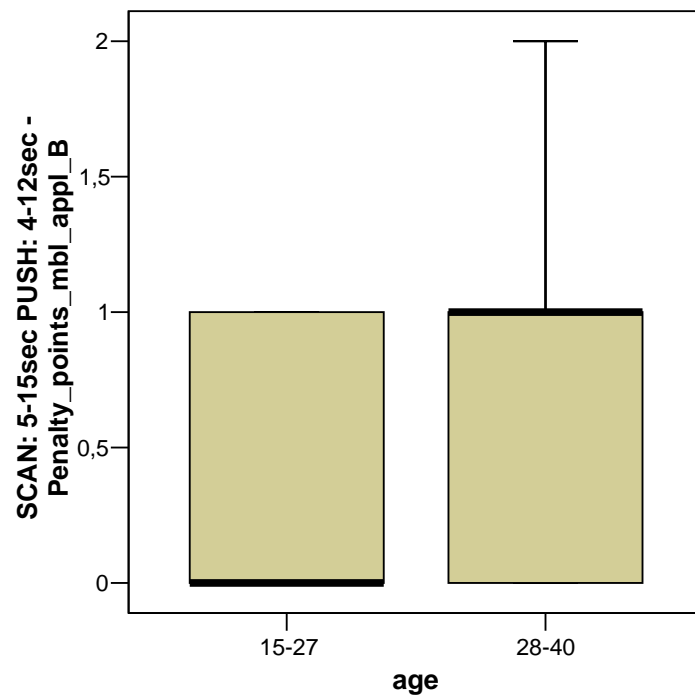


Figure 8.37 : Average Penalty_points_mbl, mbl_appl_B~age
(shortSCAN-shortPUSH)

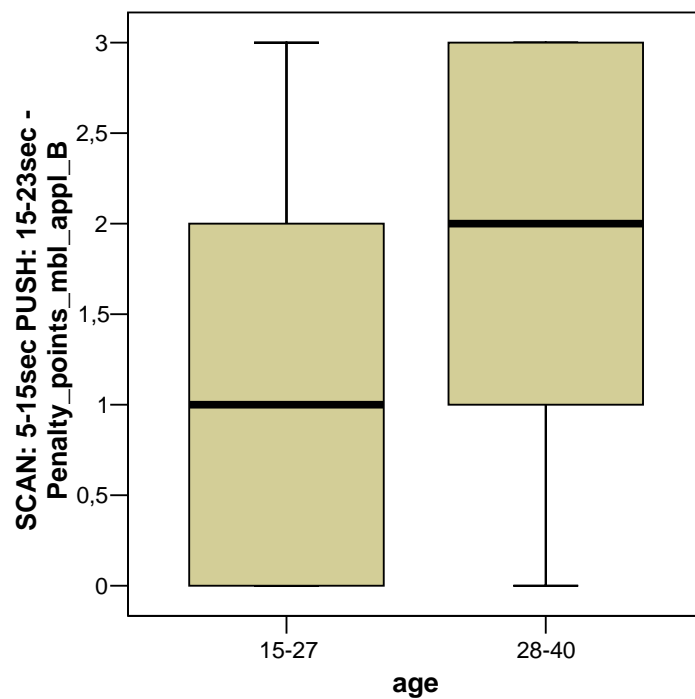


Figure 8.38 : Average Penalty_points_mbl, mbl_appl_B~age
(shortSCAN-longPUSH)

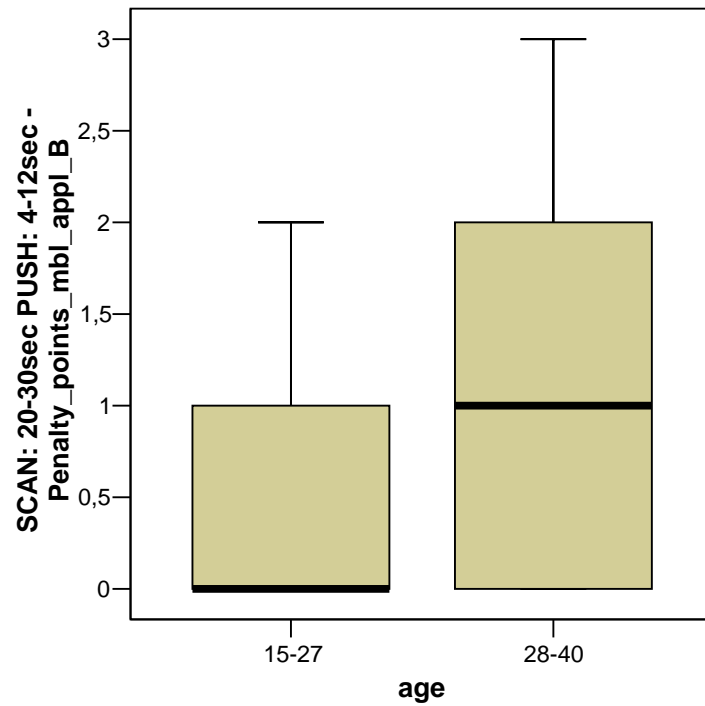


Figure 8.39 : Average Penalty_points_mbl, mbl_appl_B~age
(longSCAN-shortPUSH)

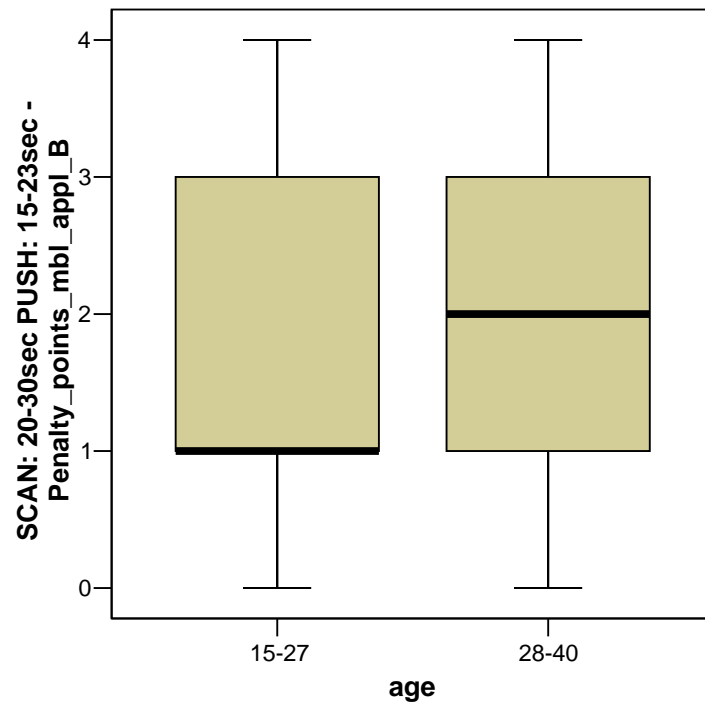


Figure 8.40 : Average Penalty_points_mbl, mbl_appl_B~age
(longSCAN-longPUSH)

The users 15-27 years old make fewer mistakes than the users 28-40 years old when the complexity of the application is higher.

8.4 Penalty Points PC

8.4.1 Mobile Application A (simple)

8.4.1.1 Scan and Push Times Graphs

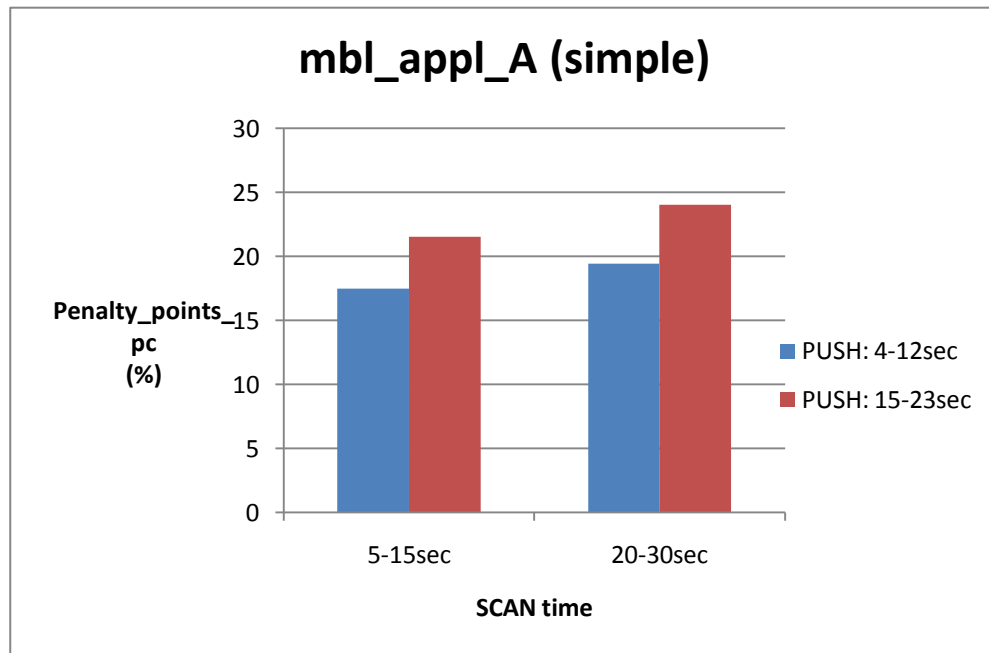


Figure 8.41 : Penalty_points_pc, mbl_appl_A (simple)~Scan time .

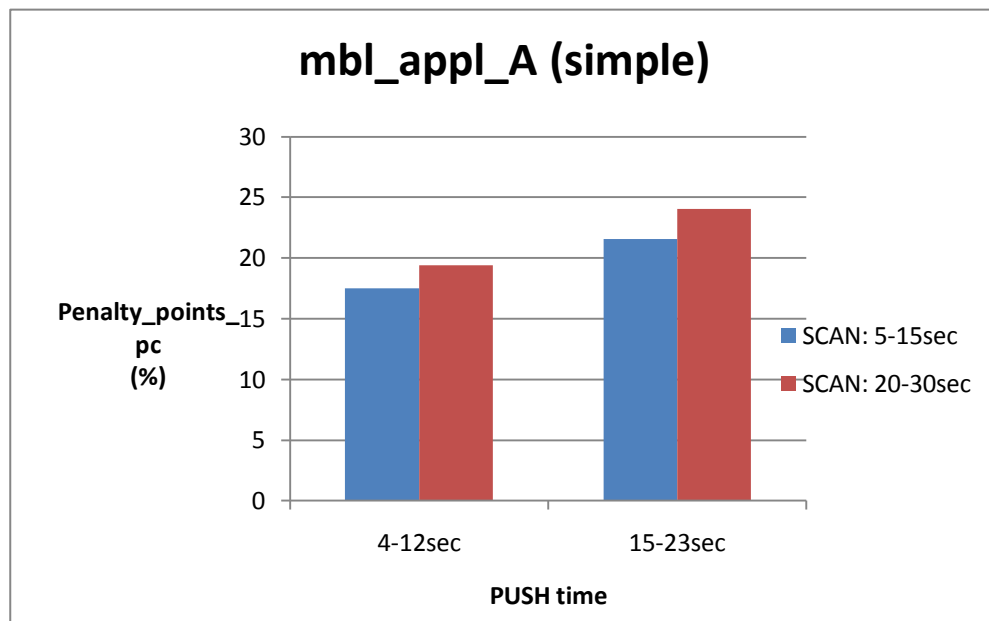


Figure 8.42 : Penalty_points_pc, mbl_appl_A (simple)~Push time .

From Figures 8.41, 8.42 we conclude that in the simple application, longer scan and push times mean more mistakes of the user as regards his external environment (this might mean that the user missed the awaited bus or hit an obstacle).

8.4.1.2 ANOVA

Within-Subjects Factors

Measure: MEASURE_1

sc_pu	Dependent Variable
1	shortSCAN_ shortPUSH
2	shortSCAN_ longPUSH
3	longSCAN_ shortPUSH
4	longSCAN_ longPUSH

Table 8.21 : The four combinations of scan and push times.

Between-Subjects Factors

	Value Label	N
gender 1	male	16
2	female	24
age 1	15-27	22
2	28-40	18

Table 8.22 : Gender and age of the 40 users.

Tests of Within-Subjects Effects

Measure: MEASURE_1

Source		Type III Sum of Squares	df	Mean Square	F	Sig.
sc_pu	Sphericity Assumed	828,672	3	276,224	43,093	,000
	Greenhouse-Geisser	828,672	1,510	548,880	43,093	,000
	Huynh-Feldt	828,672	1,693	489,486	43,093	,000
	Lower-bound	828,672	1,000	828,672	43,093	,000
sc_pu * gender	Sphericity Assumed	2,968	3	,989	,154	,927
	Greenhouse-Geisser	2,968	1,510	1,966	,154	,797
	Huynh-Feldt	2,968	1,693	1,753	,154	,822
	Lower-bound	2,968	1,000	2,968	,154	,697
sc_pu * age	Sphericity Assumed	12,926	3	4,309	,672	,571
	Greenhouse-Geisser	12,926	1,510	8,561	,672	,475
	Huynh-Feldt	12,926	1,693	7,635	,672	,491
	Lower-bound	12,926	1,000	12,926	,672	,418
sc_pu * gender * age	Sphericity Assumed	12,726	3	4,242	,662	,577
	Greenhouse-Geisser	12,726	1,510	8,429	,662	,479
	Huynh-Feldt	12,726	1,693	7,517	,662	,495
	Lower-bound	12,726	1,000	12,726	,662	,421
Error(sc_pu)	Sphericity Assumed	692,281	108	6,410		
	Greenhouse-Geisser	692,281	54,351	12,737		
	Huynh-Feldt	692,281	60,946	11,359		
	Lower-bound	692,281	36,000	19,230		

Table 8.23 : Penalty_points_pc, mbl_appl_A (simple), ANOVA table, Tests of Within-Subjects Effects.

Tests of Between-Subjects Effects

Measure: MEASURE_1

Transformed Variable: Average

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Intercept	61889,694	1	61889,694	419,091	,000
gender	172,172	1	172,172	1,166	,287
age	200,756	1	200,756	1,359	,251
gender * age	31,000	1	31,000	,210	,650
Error	5316,342	36	147,676		

Table 8.24 : Penalty_points_pc, mbl_appl_A (simple), ANOVA table, Tests of Between-Subjects Effects.

From Table 8.23 we take the following results:

- Influence of the factor sc_pu (= combination of scan and push time)
 $F(3,108) = 43,093$ $p = 0,000 < 0,001 < 0,05 \Rightarrow$ statistically significant
- Influence of the mutual influence of the factors sc_pu and gender
 $F(3,108) = 0,154$ $p = 0,927 > 0,05 \Rightarrow$ statistically insignificant
- Influence of the mutual influence of the factors sc_pu and age
 $F(3,108) = 0,672$ $p = 0,571 > 0,05 \Rightarrow$ statistically insignificant
- Influence of the mutual influence of the factors sc_pu, gender and age
 $F(3,108) = 0,662$ $p = 0,577 > 0,05 \Rightarrow$ statistically insignificant

From Table 8.24 we take the following results:

- Influence of the factor gender
 $F(1,36) = 1,166$ $p = 0,287 > 0,05 \Rightarrow$ statistically insignificant
- Influence of the factor age
 $F(1,36) = 1,359$ $p = 0,251 > 0,05 \Rightarrow$ statistically insignificant

We conclude that in the simple application, the combination of scan and push time affects very much the mistakes that each user makes as regards the external environment. The gender and the age of each user do not affect the penalty points.

8.4.1.3 Gender and Age Graphs

gender	age		SCAN: 5-15sec PUSH: 4-12sec Penalty_point s_pc_A	SCAN: 5-15sec PUSH: 15-23sec Penalty_point s_pc_A	SCAN: 20-30sec PUSH: 4-12sec Penalty_point s_pc_A	SCAN: 20-30sec PUSH: 15-23sec Penalty_point s_pc_A
male	15-27	Mean	14,9014	17,7986	16,6986	21,5386
		N	7	7	7	7
		Std. Deviation	3,82377	4,23012	3,63570	5,38881
		Minimum	9,23	10,10	9,82	13,74
		Maximum	20,51	21,92	20,95	27,94
		Range	11,28	11,82	11,13	14,20
		Median	14,2900	17,7800	17,3500	21,0100
	28-40	Mean	17,8678	22,1167	20,1244	23,7967
		N	9	9	9	9
		Std. Deviation	7,87314	6,42410	6,16994	6,43566
		Minimum	5,36	11,22	9,90	14,49
		Maximum	31,75	34,78	29,46	37,24
		Range	26,39	23,56	19,56	22,75
		Median	17,1900	21,1800	21,3700	23,3600
	Total	Mean	16,5700	20,2275	18,6256	22,8087
		N	16	16	16	16
		Std. Deviation	6,42010	5,83629	5,35455	5,91977
		Minimum	5,36	10,10	9,82	13,74
		Maximum	31,75	34,78	29,46	37,24
		Range	26,39	24,68	19,64	23,50
		Median	15,7250	20,7900	18,7200	22,3400

female	15-27	Mean	17,0827	22,1433	19,2727	24,7187
		N	15	15	15	15
		Std. Deviation	6,73388	8,37074	6,84185	9,51690
		Minimum	4,35	5,66	4,17	6,06
		Maximum	32,00	38,68	31,73	45,65
		Range	27,65	33,02	27,56	39,59
		Median	15,5200	19,5700	17,8200	21,8300
	28-40	Mean	19,7622	22,8856	21,1189	25,1022
		N	9	9	9	9
		Std. Deviation	3,90610	4,11529	3,94978	4,69776
		Minimum	14,08	16,48	15,46	17,16
		Maximum	24,62	26,97	25,69	31,65
		Range	10,54	10,49	10,23	14,49
		Median	21,4300	23,4700	21,0100	25,5600
	Total	Mean	18,0875	22,4217	19,9650	24,8625
		N	24	24	24	24
		Std. Deviation	5,88765	6,97683	5,89521	7,92733
		Minimum	4,35	5,66	4,17	6,06
		Maximum	32,00	38,68	31,73	45,65
		Range	27,65	33,02	27,56	39,59
		Median	16,3800	22,0400	18,8000	24,3350
Total	15-27	Mean	16,3886	20,7609	18,4536	23,7068
		N	22	22	22	22
		Std. Deviation	5,95726	7,49103	6,04068	8,42474
		Minimum	4,35	5,66	4,17	6,06
		Maximum	32,00	38,68	31,73	45,65
		Range	27,65	33,02	27,56	39,59

	28-40	Median	15,3800	19,0450	17,3700	21,5000
		Mean	18,8150	22,5011	20,6217	24,4494
		N	18	18	18	18
		Std. Deviation	6,10738	5,24852	5,05151	5,50702
		Minimum	5,36	11,22	9,90	14,49
		Maximum	31,75	34,78	29,46	37,24
		Range	26,39	23,56	19,56	22,75
		Median	18,3750	21,6950	21,1900	24,3300
	Total	Mean	17,4805	21,5440	19,4293	24,0410
		N	40	40	40	40
		Std. Deviation	6,07148	6,55686	5,65373	7,18175
		Minimum	4,35	5,66	4,17	6,06
		Maximum	32,00	38,68	31,73	45,65
		Range	27,65	33,02	27,56	39,59
		Median	16,2150	21,3050	18,8000	23,3350

Table 8.25 : Total results of Penalty_points_pc, mbl_appl_A (simple).

GENDER

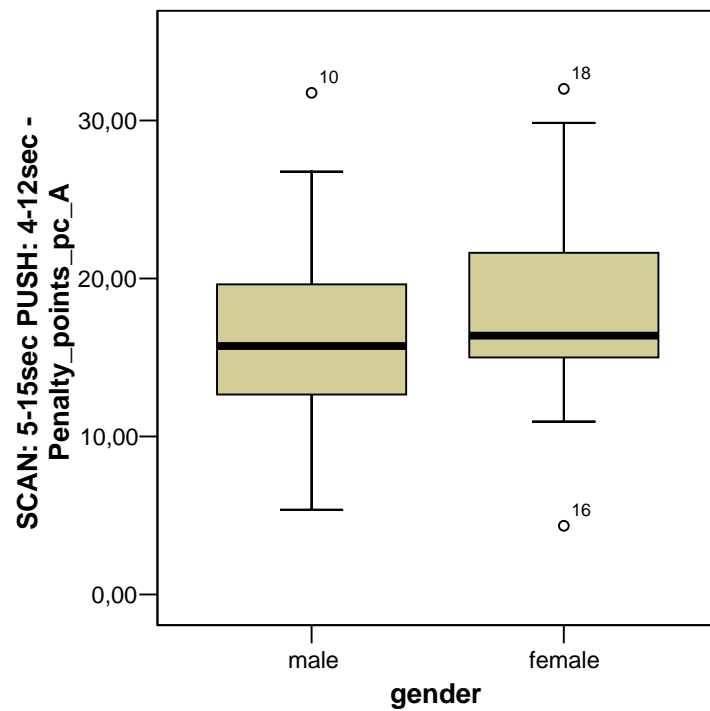


Figure 8.43 : Average Penalty_points_pc, mbl_appl_A~gender (shortSCAN-shortPUSH)

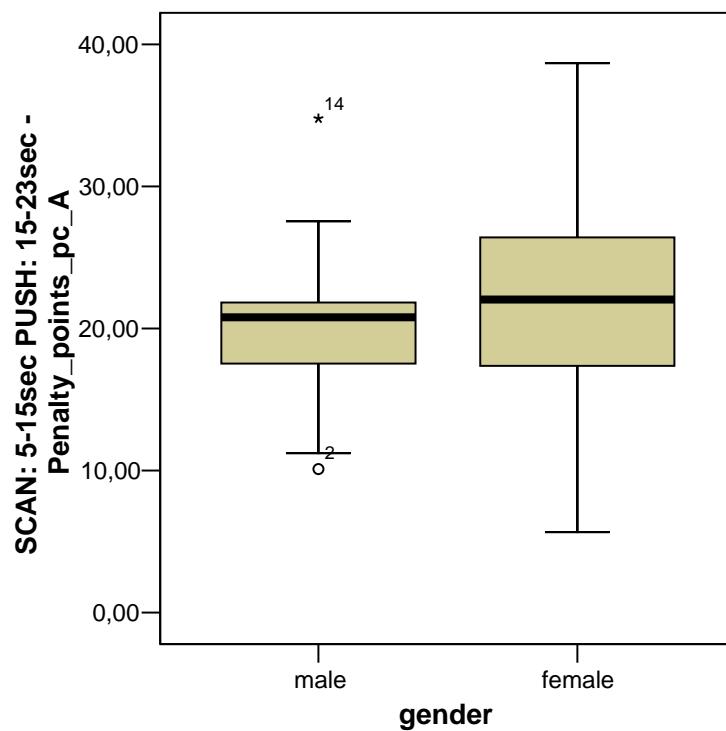


Figure 8.44 : Average Penalty_points_pc, mbl_appl_A~gender (shortSCAN-longPUSH)

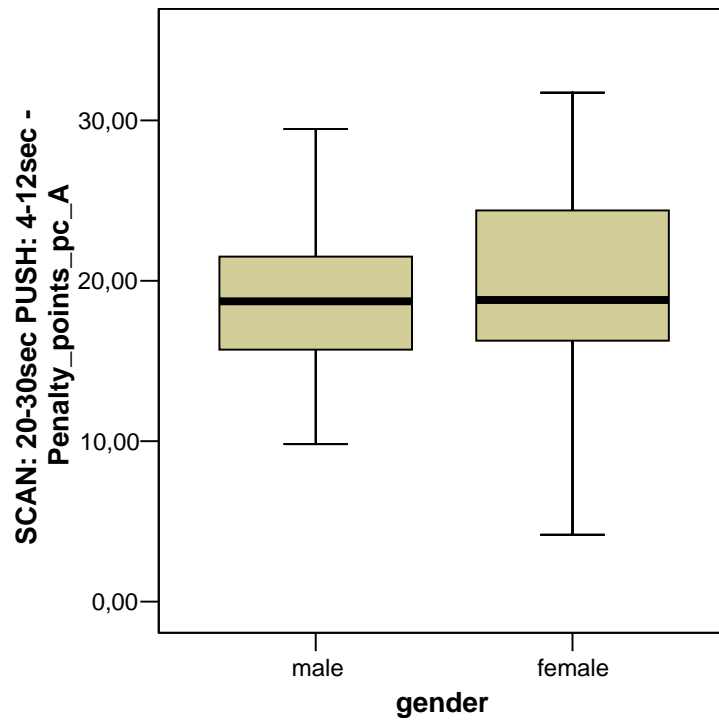


Figure 8.45 : Average Penalty_points_pc, mbl_appl_A~gender (longSCAN-shortPUSH)

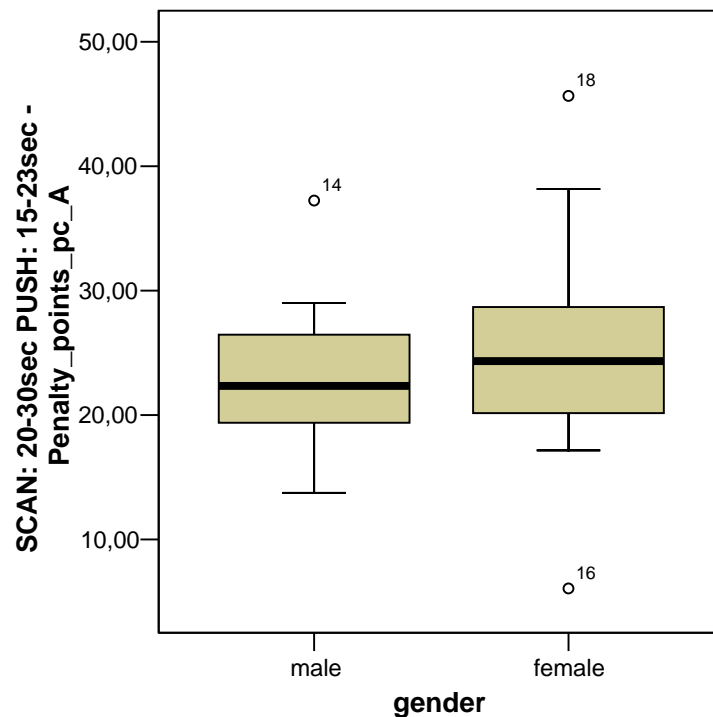


Figure 8.46 : Average Penalty_points_pc, mbl_appl_A~gender (longSCAN-longPUSH)

In the simple application, women seem to be a bit more careless to their external environment during an interactive procedure with their mobile phone.

AGE

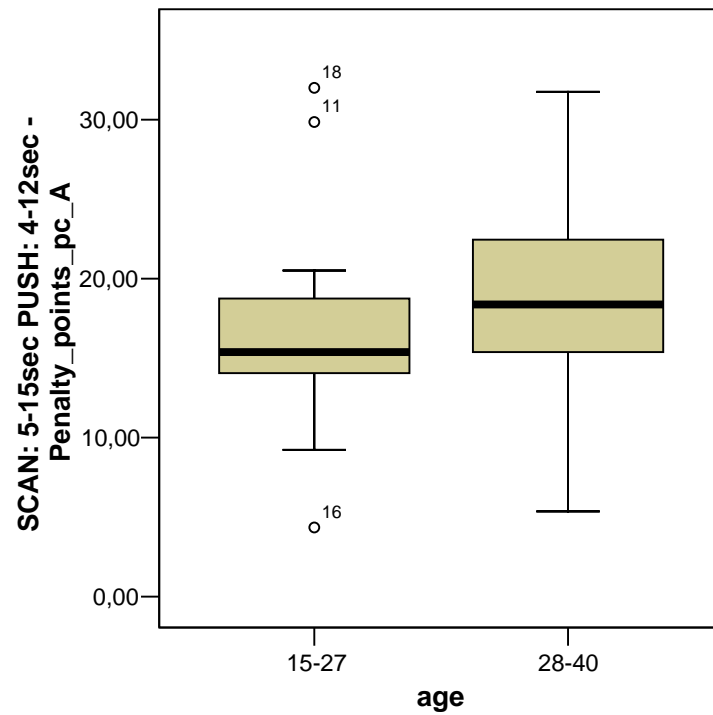


Figure 8.47 : Average Penalty_points_pc, mbl_appl_A~age (shortSCAN-shortPUSH)

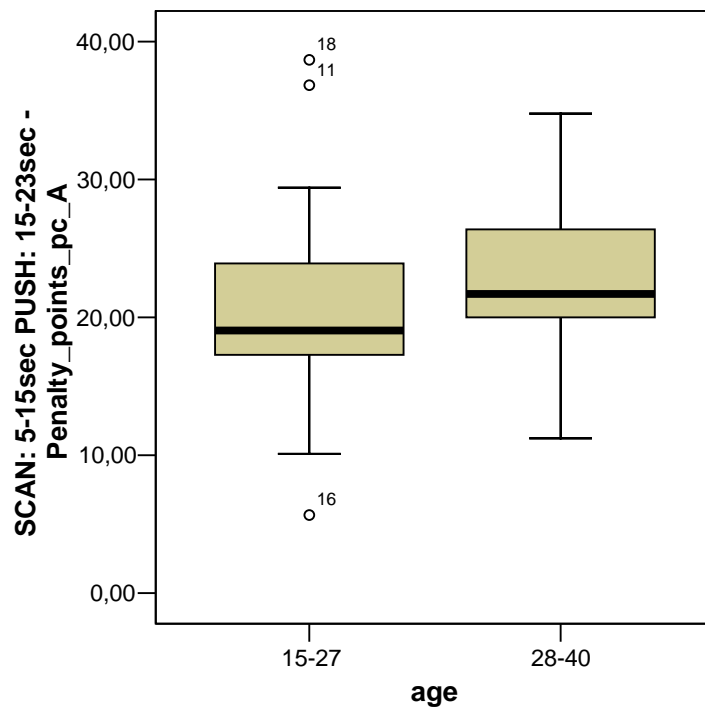


Figure 8.48 : Average Penalty_points_pc, mbl_appl_A~age (shortSCAN-longPUSH)

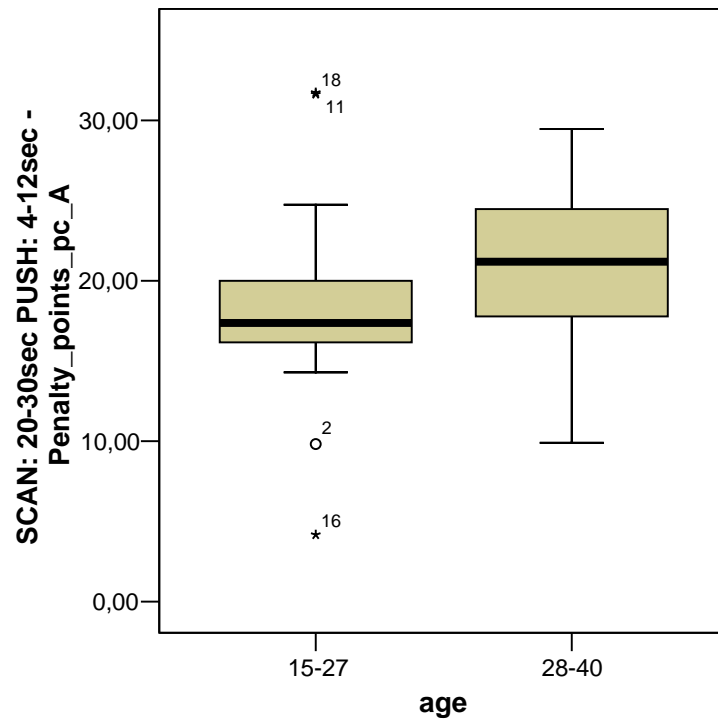


Figure 8.49 : Average Penalty_points_pc, mbl_appl_A~age
(longSCAN-shortPUSH)

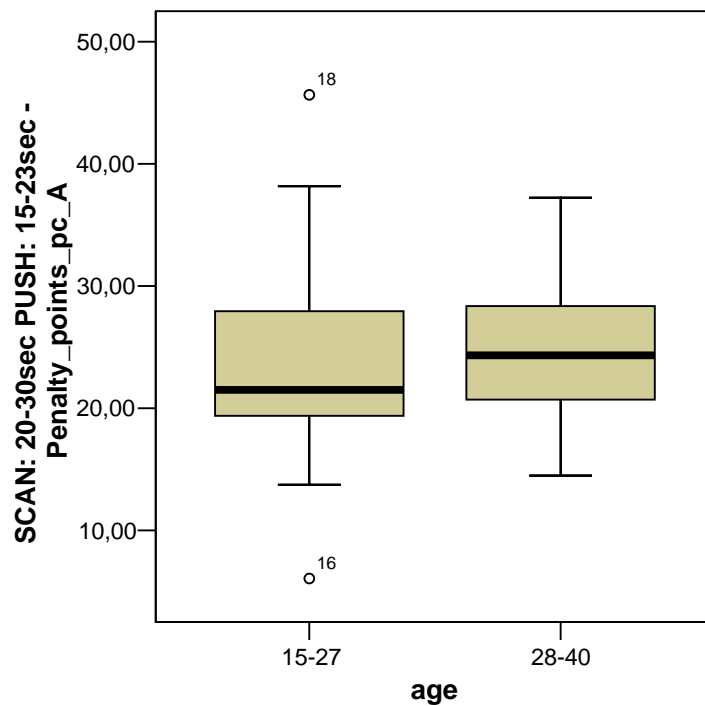


Figure 8.50 : Average Penalty_points_pc, mbl_appl_A~age
(longSCAN-longPUSH)

In the simple application the users 15-27 years old can manage an interactive procedure with their mobile phone better than the users 28-40 years old.

8.4.2 Mobile Application B (complex)

8.4.2.1 Scan and Push Times Graphs

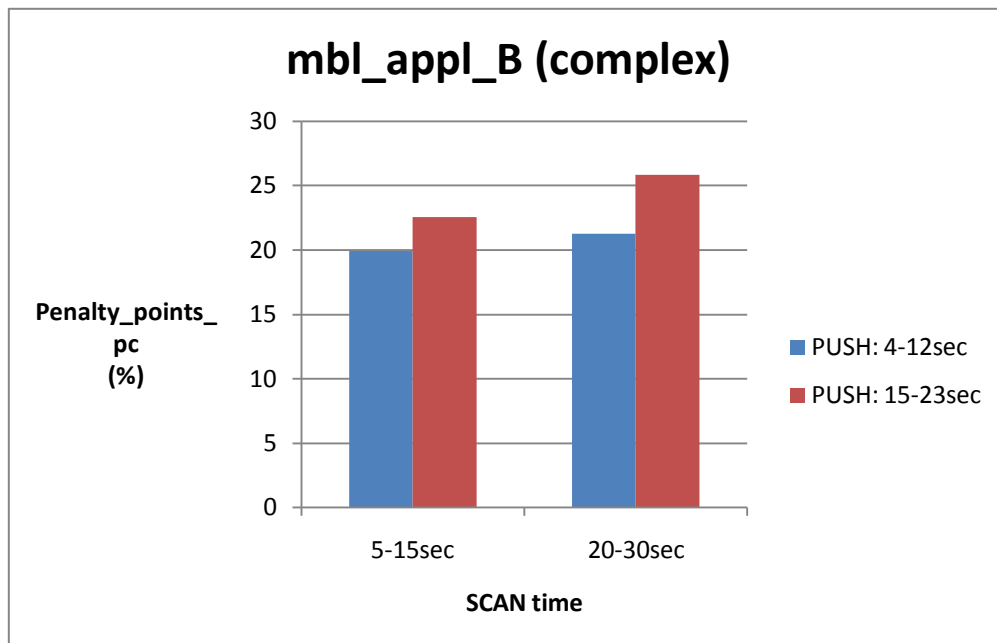


Figure 8.51 : Penalty_points_pc, mbl_appl_B (complex)~Scan time .

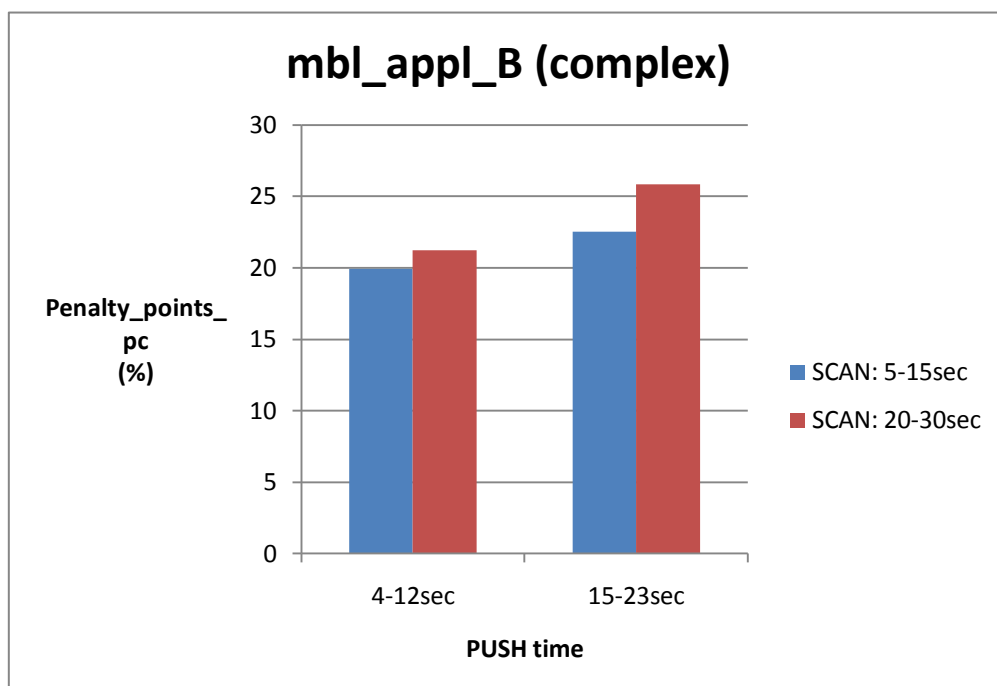


Figure 8.52 : Penalty_points_pc, mbl_appl_B (complex)~Push time .

From Figures 8.51, 8.52 we conclude that in the complex application, longer scan and push times mean more mistakes of the user as regards his external environment (this might mean that the user missed the awaited bus or hit an obstacle).

8.4.2.2 ANOVA

Within-Subjects Factors

Measure: MEASURE_1

sc pu	Dependent Variable
1	shortSCAN_ shortPUSH
2	shortSCAN_ longPUSH
3	longSCAN_ shortPUSH
4	longSCAN_ longPUSH

Table 8.26 : The four combinations of scan and push times.

Between-Subjects Factors

	Value Label	N
gender 1	male	16
2	female	24
age 1	15-27	22
2	28-40	18

Table 8.27 : Gender and age of the 40 users.

Tests of Within-Subjects Effects

Measure: MEASURE_1

Source		Type III Sum of Squares	df	Mean Square	F	Sig.
sc_pu	Sphericity Assumed	704,085	3	234,695	78,457	,000
	Greenhouse-Geisser	704,085	1,780	395,539	78,457	,000
	Huynh-Feldt	704,085	2,022	348,163	78,457	,000
	Lower-bound	704,085	1,000	704,085	78,457	,000
sc_pu * gender	Sphericity Assumed	1,761	3	,587	,196	,899
	Greenhouse-Geisser	1,761	1,780	,989	,196	,797
	Huynh-Feldt	1,761	2,022	,871	,196	,825
	Lower-bound	1,761	1,000	1,761	,196	,660
sc_pu * age	Sphericity Assumed	6,407	3	2,136	,714	,546
	Greenhouse-Geisser	6,407	1,780	3,599	,714	,478
	Huynh-Feldt	6,407	2,022	3,168	,714	,495
	Lower-bound	6,407	1,000	6,407	,714	,404
sc_pu * gender * age	Sphericity Assumed	,544	3	,181	,061	,980
	Greenhouse-Geisser	,544	1,780	,305	,061	,925
	Huynh-Feldt	,544	2,022	,269	,061	,943
	Lower-bound	,544	1,000	,544	,061	,807
Error(sc_pu)	Sphericity Assumed	323,067	108	2,991		
	Greenhouse-Geisser	323,067	64,082	5,041		
	Huynh-Feldt	323,067	72,802	4,438		
	Lower-bound	323,067	36,000	8,974		

Table 8.28 : Penalty_points_pc, mbl_appl_B (complex), ANOVA table, Tests of Within-Subjects Effects.

Tests of Between-Subjects Effects

Measure: MEASURE_1

Transformed Variable: Average

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Intercept	71855,070	1	71855,070	435,921	,000
gender	626,756	1	626,756	3,802	,059
age	51,638	1	51,638	,313	,579
gender * age	2,586	1	2,586	,016	,901
Error	5934,064	36	164,835		

Table 8.29 : Penalty_points_pc, mbl_appl_B (complex), ANOVA table, Tests of Between-Subjects Effects.

From Table 8.28 we take the following results:

- Influence of the factor sc_pu (= combination of scan and push time)
 $F(3,108) = 78,457$ $p = 0,000 < 0,001 < 0,05$ \Rightarrow statistically significant
- Influence of the mutual influence of the factors sc_pu and gender
 $F(3,108) = 0,196$ $p = 0,899 > 0,05$ \Rightarrow statistically insignificant
- Influence of the mutual influence of the factors sc_pu and age
 $F(3,108) = 0,714$ $p = 0,546 > 0,05$ \Rightarrow statistically insignificant
- Influence of the mutual influence of the factors sc_pu, gender and age
 $F(3,108) = 0,061$ $p = 0,980 > 0,05$ \Rightarrow statistically insignificant

From Table 8.29 we take the following results:

- Influence of the factor gender
 $F(1,36) = 3,802$ $p = 0,059 > 0,05$ \Rightarrow statistically insignificant
- Influence of the factor age
 $F(1,36) = 0,313$ $p = 0,579 > 0,05$ \Rightarrow statistically insignificant

We conclude that in the complex application, the combination of scan and push time affects very much the mistakes that each user makes as regards the external environment. The gender and the age of each user do not affect the penalty points.

8.4.2.3 Gender and Age Graphs

gender	age		SCAN: 5-15sec PUSH: 4-12sec Penalty_point s_pc_B	SCAN: 5-15sec PUSH: 15-23sec Penalty_point s_pc_B	SCAN: 20-30sec PUSH: 4-12sec Penalty_point s_pc_B	SCAN: 20-30sec PUSH: 15-23sec Penalty_point s_pc_B
male	15-27	Mean	16,9471	19,2329	18,4600	22,3129
		N	7	7	7	7
		Std. Deviation	6,24546	6,17671	5,52035	5,71281
		Minimum	8,93	11,36	10,81	14,39
		Maximum	24,62	26,80	25,44	30,43
		Range	15,69	15,44	14,63	16,04
		Median	17,2400	20,6500	18,4500	21,3800
	28-40	Mean	18,5133	20,7456	19,1078	24,3644
		N	9	9	9	9
		Std. Deviation	5,63498	6,04037	5,74295	5,47182
		Minimum	11,11	12,15	10,48	12,86
		Maximum	29,33	30,34	29,59	31,54
		Range	18,22	18,19	19,11	18,68
		Median	19,0500	21,9800	18,3700	24,6300
	Total	Mean	17,8281	20,0837	18,8244	23,4669
		N	16	16	16	16
		Std. Deviation	5,76031	5,94312	5,46717	5,48888
		Minimum	8,93	11,36	10,48	12,86
		Maximum	29,33	30,34	29,59	31,54
		Range	20,40	18,98	19,11	18,68
		Median	19,0100	20,8800	18,4100	23,6850

female	15-27	Mean	20,8727	23,8407	22,7447	27,0013
		N	15	15	15	15
		Std. Deviation	7,09166	7,79163	7,54292	7,86225
		Minimum	7,69	12,50	11,32	14,29
		Maximum	35,94	40,00	36,94	43,88
		Range	28,25	27,50	25,62	29,59
		Median	19,3500	22,1100	20,5400	25,5500
	28-40	Mean	22,0989	24,8022	23,0856	28,1378
		N	9	9	9	9
		Std. Deviation	5,38684	5,90344	6,59257	6,03760
		Minimum	15,25	15,56	12,50	17,16
		Maximum	31,34	36,17	34,95	37,21
		Range	16,09	20,61	22,45	20,05
		Median	22,0600	25,5100	23,7100	29,0100
	Total	Mean	21,3325	24,2012	22,8725	27,4275
		N	24	24	24	24
		Std. Deviation	6,40884	7,02151	7,05534	7,11488
		Minimum	7,69	12,50	11,32	14,29
		Maximum	35,94	40,00	36,94	43,88
		Range	28,25	27,50	25,62	29,59
		Median	21,0150	24,3000	22,4800	28,6650
Total	15-27	Mean	19,6236	22,3745	21,3814	25,5095
		N	22	22	22	22
		Std. Deviation	6,94078	7,49659	7,12810	7,45188
		Minimum	7,69	11,36	10,81	14,29
		Maximum	35,94	40,00	36,94	43,88
		Range	28,25	28,64	26,13	29,59

	28-40	Median	19,0500	21,5050	20,2900	23,6200
		Mean	20,3061	22,7739	21,0967	26,2511
		N	18	18	18	18
		Std. Deviation	5,65697	6,15844	6,33733	5,91717
		Minimum	11,11	12,15	10,48	12,86
		Maximum	31,34	36,17	34,95	37,21
		Range	20,23	24,02	24,47	24,35
		Median	20,2750	23,8350	21,7100	26,9850
	Total	Mean	19,9307	22,5542	21,2533	25,8433
		N	40	40	40	40
		Std. Deviation	6,32516	6,84350	6,69971	6,73072
		Minimum	7,69	11,36	10,48	12,86
		Maximum	35,94	40,00	36,94	43,88
		Range	28,25	28,64	26,46	31,02
		Median	19,2350	22,0450	20,4650	25,8200

Table 8.30 : Total results of Penalty_points_pc, mbl_appl_B (complex).

GENDER

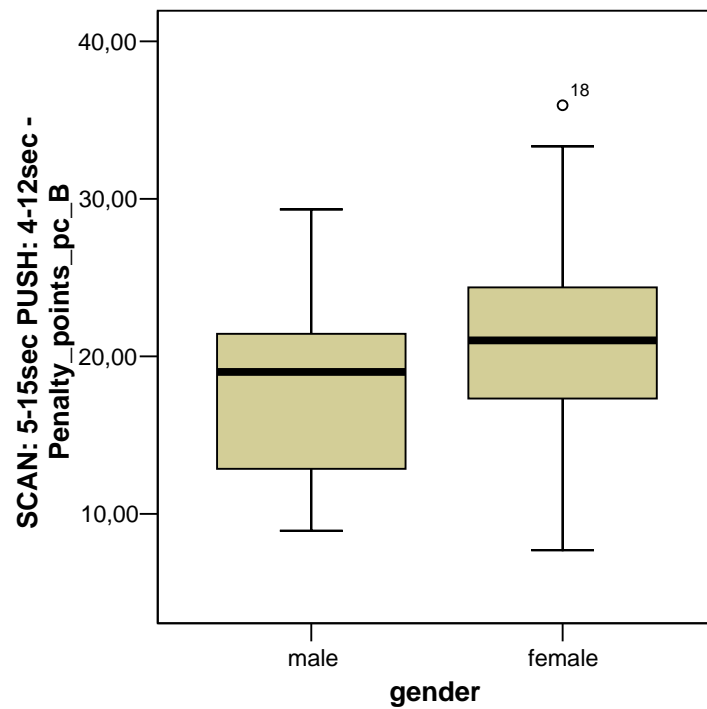


Figure 8.53 : Average Penalty_points_pc, mbl_appl_B~gender (shortSCAN-shortPUSH)

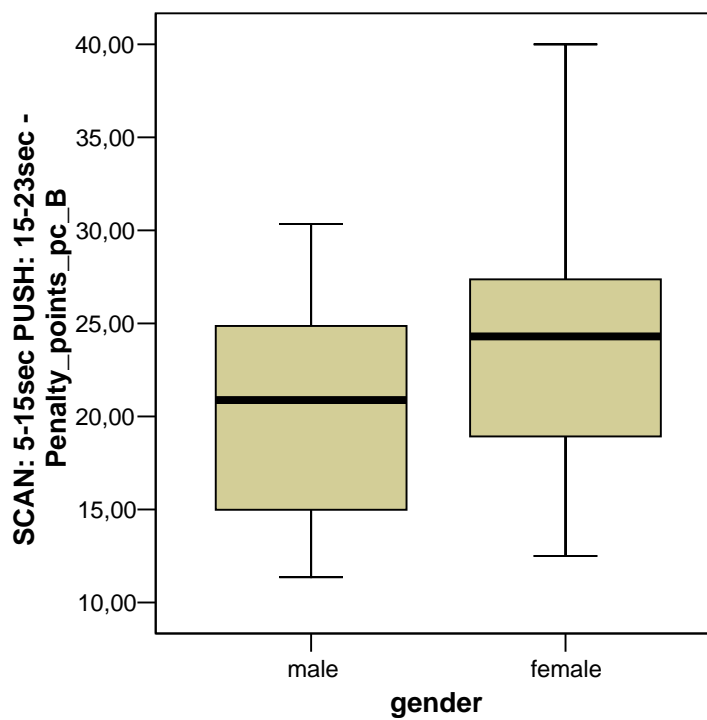


Figure 8.54 : Average Penalty_points_pc, mbl_appl_B~gender (shortSCAN-longPUSH)

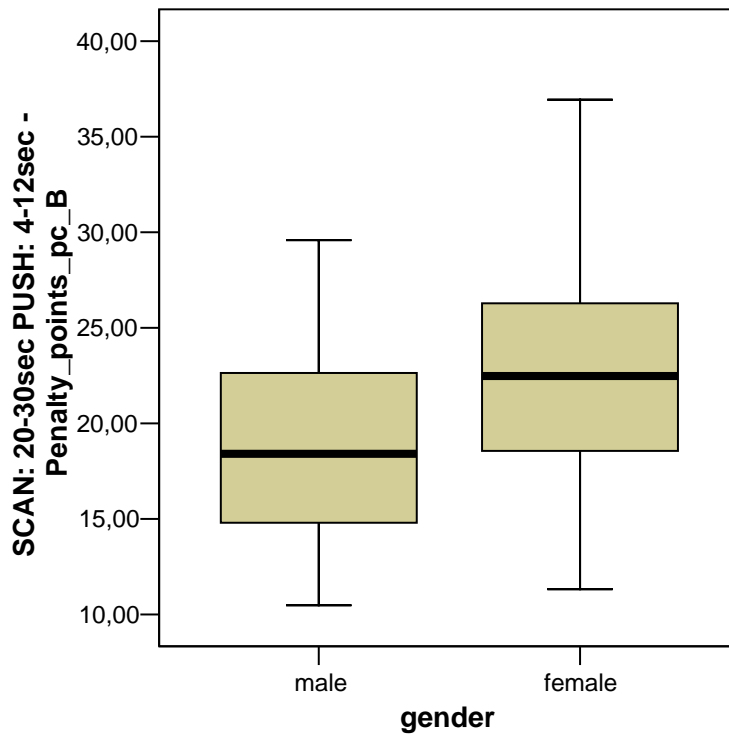


Figure 8.55 : Average Penalty_points_pc, mbl_appl_B~gender (longSCAN-shortPUSH)

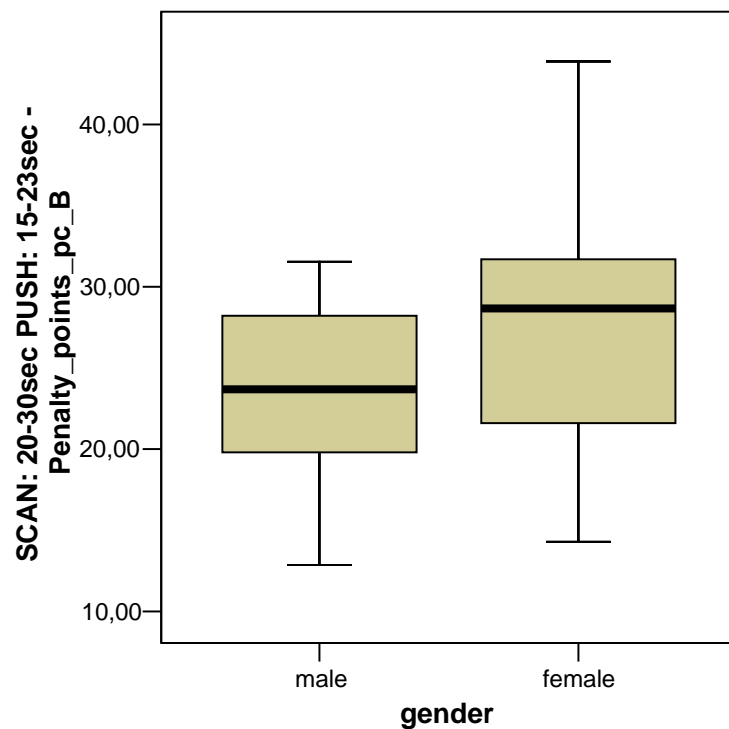


Figure 8.56 : Average Penalty_points_pc, mbl_appl_B~gender (longSCAN-longPUSH)

In the complex application, women seem to be more careless to their external environment during an interactive procedure with their mobile phone.

AGE

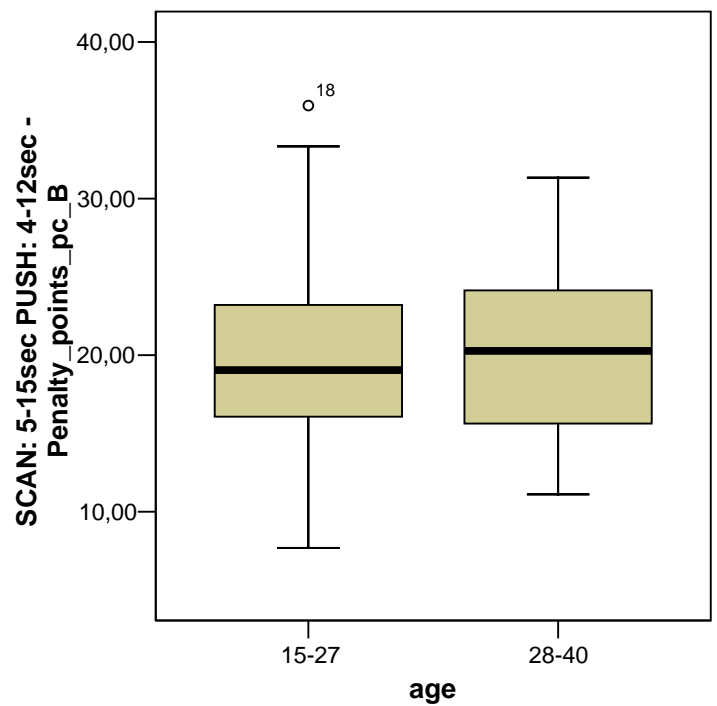


Figure 8.57 : Average Penalty_points_pc, mbl_appl_B~age (shortSCAN-shortPUSH)

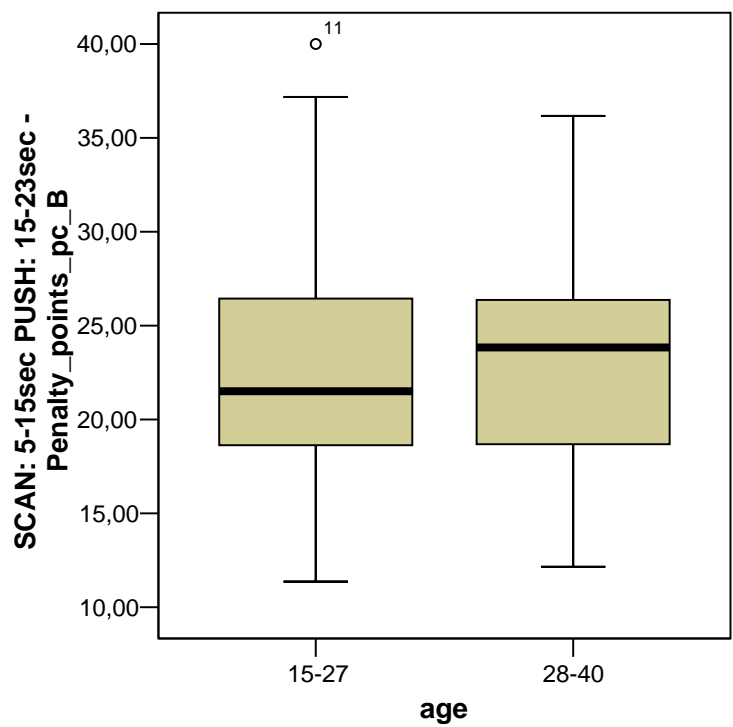


Figure 8.58 : Average Penalty_points_pc, mbl_appl_B~age (shortSCAN-longPUSH)

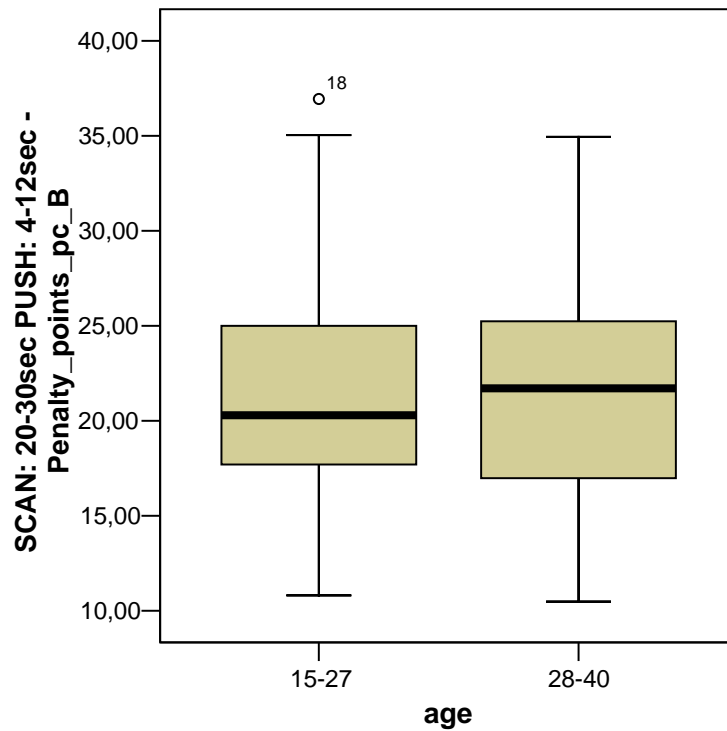


Figure 8.59 : Average Penalty_points_pc, mbl_appl_B~age (longSCAN-shortPUSH)

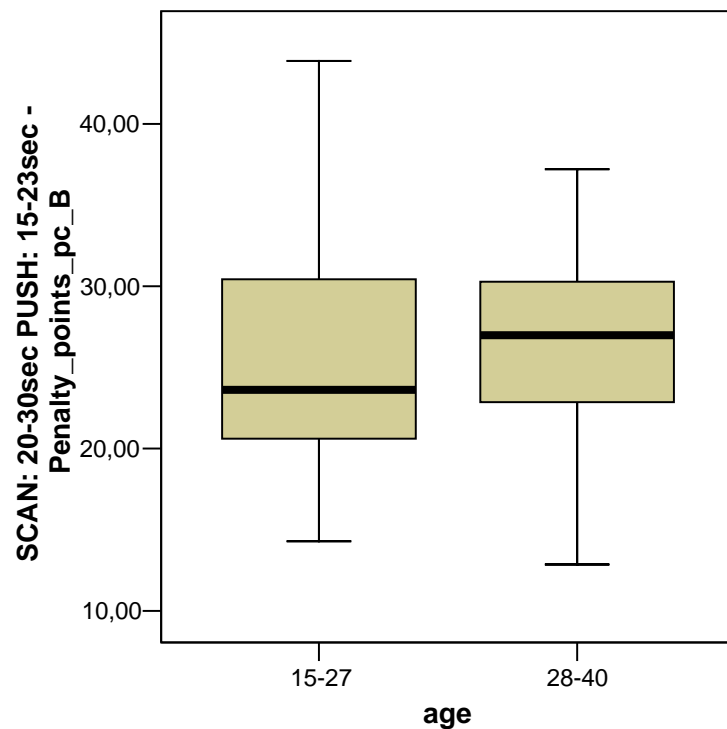


Figure 8.60 : Average Penalty_points_pc, mbl_appl_B~age (longSCAN-longPUSH)

In the complex application the users 15-27 years old can manage an interactive procedure with their mobile phone in a better way than the users 28-40 years old.

8.5 Total Results and Conclusions

8.5.1 Mobile Application A (simple)

mbl_appl_A (simple)				
	SCAN : 5-15sec PUSH : 4-12sec	SCAN : 5-15sec PUSH : 15-23sec	SCAN : 20-30sec PUSH : 4-12sec	SCAN : 20-30sec PUSH : 15-23sec
Duration on Average (sec)	18	29	33	44
Response time (sec)	2385,40	3147,63	2755,13	3826,48
Penalty Points mbl	0,10	0,57	0,28	1
Penalty Points pc (%)	17,48	21,54	19,43	24,04

Table 8.31 : Total_results, mbl_appl_A (simple)

8.5.2 Mobile Application B (complex)

mbl_appl_B (complex)				
	SCAN : 5-15sec PUSH : 4-12sec	SCAN : 5-15sec PUSH : 15-23sec	SCAN : 20-30sec PUSH : 4-12sec	SCAN : 20-30sec PUSH : 15-23sec
Duration on Average (sec)	18	29	33	44
Response time (sec)	3166,23	3938,15	3569,30	4628,23
Penalty Points mbl	0,55	1,20	0,72	1,78
Penalty Points pc (%)	19,93	22,55	21,25	25,84

Table 8.32 : Total_results, mbl_appl_B (complex)

8.5.3 Conclusions

In the tables above we calculated the duration of each scan and push time combination on average. The longer this duration is, the more mistakes the user makes and the longer his response time is. What is interesting and we should highlight is the exception that appears in the red columns. The reason for this is the push time. Although the average duration in the first red column is shorter, the values of our measurements are higher. But push time is longer in the first red column than the one in the second. The conclusion is that the duration of push time affects the user more than the scan time and should be kept in low values in order to make the whole procedure effective and not annoying for the user.

9

REFERENCES

- [1] Anastasi, G., Bandelloni, R., Conti, M., Delmastro, F., Gregori, E. & Mainetto, G. (2003) "Experimenting an Indoor Bluetooth-Based Positioning Service", Engineering department, University of Pisa, Proceedings of the 23rd International Conference on Distributed Computing Systems.
- [2] Lauri A., Gothlin N., Korhonen J., Ojala T., (2004) "Bluetooth and WAP Push Based Location-Aware Mobile Advertising System", International Conference On Mobile Systems, Applications And Services, Proceedings of the 2nd international conference on Mobile systems, applications, and services. Session: Mobile Applications.
- [3] Erik Welsh, Patrick Murphy, J. Patrick Frantz, "Improving Connection Times for Bluetooth Devices in Mobile Environments", Rice University
- [4] Marie Duflot, Marta Kwiatkowska, Gethin Norman, David Parker, "A Formal Analysis of Bluetooth Device Discovery", International Journal on Software Tools for Technology Transfer (STTT)
- [5] Barwise P., Strong C. (2002) "Permission-Based Mobile Advertising", Journal of Interactive Marketing. Vol. 16, no. 1
- [6] Ranganathan A., Campbell R. (2002) "Advertising in a Pervasive Computing Environment". International Conference on Mobile Computing and Networking. 2nd International Workshop on Mobile Commerce.

- [7] Yunos H., Gao J., Shim S., (2003) "Wireless Advertising's Challenges and Opportunities". IEEE Computer. Vol. 36, no. 5
- [8] Leppaniemi M., Karjaluto H., (2005) "Factors influencing consumer's willingness to accept mobile advertising : a conceptual model". International Journal Mobile Communications, Vol. 3, no. 3
- [9] Melody M Tsang, Shu-Chun Ho, Ting-Peng Liang, "Consumer Attitudes Toward Mobile Advertising: An Empirical Study".
- [10] Salo J., Tahtinen J., "Retailer Use of Permission-Based Mobile Advertising", Department of Marketing, University of Oulu, Finland.
- [11] H. Toutenburg, "Statistical Analysis of Designed Experiments", Second Edition, Springer, 2002.
- [12] Chugunov A., Karppinen J, "Personalisation in Mobile Advertising Systems", Oulu Polytechnic/ Pehr Brahe Software Laboratory (PBOL), Finland.
- [13] Varshney, U. Vetter, R, "A framework for the emerging mobile commerce applications", Dept. of Comput. Inf. Syst., Georgia State Univ., Atlanta, GA, USA System Sciences, 2001, Proceedings of the 34th Annual Hawaii International Conference.
- [14] J. Bray, C.F. Sturman, "Bluetooth 1.1 Connect without Cables", Second edition, Prentice Hall PTR, 2002.
- [15] J. Rick Turner, Julian F. Thayer, "Introduction to ANALYSIS of VARIANCE", Sage Publications, 2001.
- [16] R. Christensen, "Analysis of Variance, Design and Regression", Chapman & Hall/CRC, 1996.

[17] Γ. Σιώμκος, Αικ. Βασιλικοπούλου, «Εφαρμογή μεθόδων ανάλυσης στην έρευνα αγοράς», Εκδόσεις Σταμούλη Α.Ε., 2005.

[18] V.I. Koshelev, “Ultra-Wideband, Short-Pulse Electromagnetics 5”, Springer US, 2002.

[19] Web site for Bluetooth <http://www.bluetooth.com>

[20] <http://www.wikipedia.org/>

[21] <http://www.plaisio.gr/>