



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ ΗΛΕΚΤΡΟΝΙΚΗΣ
ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

**Μεταφορά και αναπαραγωγή κινούμενης εικόνας και ήχου (video)
πραγματικού χρόνου για κινητές συσκευές πάνω από δίκτυα
Bluetooth**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ
των

Αλέξιος Δ. Σουραβλής

Χαράλαμπος Γ. Χαραλαμπίδης

Επιβλέπων : Ε. Ν. Πρωτονοτάριος
Καθηγητής Ε.Μ.Π

Συνεπιβλέπων: Δρ. Χ. Ζ. Πατρικάκης

Αθήνα, Ιούλιος 2004



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ ΗΛΕΚΤΡΟΝΙΚΗΣ
ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

**Μεταφορά και αναπαραγωγή κινούμενης εικόνας και ήχου (video)
πραγματικού χρόνου για κινητές συσκευές πάνω από δίκτυα
Bluetooth**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Αλέξιος Δ. Σουραβλής

Χαράλαμπος Γ. Χαραλαμπίδης

Επιβλέπων : Ε.Ν. Πρωτονοτάριος
Καθηγητής Ε.Μ.Π

Συνεπιβλέπων: Δρ. Χ. Ζ. Πατρικάκης

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 22^η Ιουλίου 2004.

.....
Ον/μο Μέλος Δ.Ε.Π
Ιδιότητα Μέλους Δ.Ε.Π

.....
Ον/μο Μέλος Δ.Ε.Π
Ιδιότητα Μέλους Δ.Ε.Π

.....
Ον/μο Μέλος Δ.Ε.Π
Ιδιότητα Μέλους Δ.Ε.Π

Αθήνα, Ιούλιος 2004

.....
Αλέξιος Δ. Σουραβλής

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

.....
Χαράλαμπος Γ. Χαραλαμπίδης

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

Copyright © Αλέξιος Σουραβλής, 2004

Copyright © Χαράλαμπος Χαραλαμπίδης, 2004

Με επιφύλαξη παντός δικαιώματος. 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.

Πρόλογος

Η παρούσα διπλωματική εργασία έχει θέμα την «Μεταφορά και αναπαραγωγή κινούμενης εικόνας και ήχου (video) πραγματικού χρόνου για κινητές συσκευές πάνω από δίκτυα Bluetooth».

Μετά την ολοκλήρωση της εργασίας αισθανόμαστε την ανάγκη να ευχαριστήσουμε τον κ. Ε.Ν. Πρωτονοτάριο, Καθηγητή ΕΜΠ, ο οποίος μας εμπιστεύθηκε την εκπόνηση της, τον κ. **Χαράλαμπο Πατρικάκη** για την πολύτιμη βοήθειά του, την καθοδήγηση και τις χρήσιμες υποδείξεις του σε όλα τα στάδια της παρούσας διπλωματικής, όλα τα μέλη του εργαστηρίου των τηλεπικοινωνιών για την υπομονή και προθυμότητα που επέδειξαν καθώς και όλους όσους μας βοήθησαν στην ολοκλήρωση αυτής της διπλωματικής εργασίας.

Τέλος, θα θέλαμε να ευχαριστήσουμε τις οικογένειές μας για τη συνεχή συμπαράστασή τους κατά την εκπόνηση της διπλωματικής εργασίας.

Περίληψη

Αντικείμενο της παρούσας διπλωματικής είναι η παρουσίαση των τεχνολογιών Bluetooth, Linux και Affix και η ανάπτυξη ενός σεναρίου χρήσης τους. Το Bluetooth πρόκειται για μια χαμηλό-τροφοδοτούμενη και φτηνή ράδιο-ζεύξη που μπορεί να χρησιμοποιηθεί για την επικοινωνία απομακρυσμένων συσκευών. Το Affix είναι μία ολοκληρωμένη και προσεκτικά υλοποιημένη εφαρμογή που έχει ως στόχο την πολύπλευρη επικοινωνία με Bluetooth συσκευές.

Στα πλαίσια του παραπάνω σεναρίου χρήσης, αναπτύξαμε μια πλατφόρμα στην οποία γίνεται εφικτή η περιήγηση στο διαδίκτυο μέσω κινητού τηλεφώνου, χωρίς τη χρήση GPRS υπηρεσιών, πετυχαίνοντας μεγάλες ταχύτητες και χωρίς χρέωση περιήγηση. Επίσης με τη χρήση κατάλληλων εφαρμογών, γίνεται ικανή η προβολή Streaming Video από κινητό τηλέφωνο.

Η εφαρμογή στο Linux είναι υπεύθυνη για την επικοινωνία με τις απομακρυσμένες συσκευές και παράλληλα διαχειρίζεται τα αρχεία video που πρόκειται να μεταφερθούν μέσω Bluetooth σε αυτές, με την βοήθεια του Affix.

Η εφαρμογές που χρησιμοποιήθηκαν στην πλευρά της έξυπνης συσκευής κινητής τηλεφωνίας, μπορούν να αναπαράγουν τα αρχεία video. Η υλοποίηση έγινε χρησιμοποιώντας την εφαρμογή Gnubox, που βασίζεται στην πλατφόρμα Series60, που παρέχεται από την εταιρία Nokia και ενός Video Player, στην περίπτωση μας τον Real One Player.

Λέξεις Κλειδιά: Affix, Linux, Gnubox, Bluetooth, Ασύρματη επικοινωνία, Nokia, HCI, L2CAP, LMP, SDP, OBEX, btctl, btsrv, echo, pppd, streaming video, κινούμενη εικόνα πραγματικού χρόνου, audio, video.

Abstract

The objective of our diploma thesis is the presentation of Bluetooth, LINUX and Affix and the implementation of a possible scenario that uses them. Bluetooth is a low-power and cheap radio link that can be used for the communication of remote devices. Affix is a Bluetooth Protocol Stack for Linux developed by Nokia Research Center in Helsinki and supports core Bluetooth protocols like HCI, L2CAP 1.1, L2CAP 1.2, RFCOMM, SDP and various Bluetooth profiles.

While trying to implement a scenario of use, we developed an application-server (in Linux environment), where a client can surf the network via Bluetooth without using GPRS services and succeeded great speeds and without charging his mobile device. Also, in this scenario the user can watch streaming video through a Smart phone.

The Linux application is in charge for the communication with the remote devices and at the same time manages the video files that are to be transported via Bluetooth, based on Affix.

The application that was developed in the smart phone can accept and play the video files. The development was made using Gnubox, which is an application, which is based on the Series60 platform, that is provided by the company Nokia and constitutes an extension of operating system Symbian OS with certain moreover possibilities with regard to the user interface and the real one player for watching streaming video.

Key Words: Affix, Linux, Gnubox, Bluetooth, Wireless Communication, HCI, L2CAP, LMP, SDP, OBEX, btctl, btsrv, echo, pppd, streaming video, audio.

| | |
|--|----|
| Πρόλογος | 5 |
| Περίληψη | 6 |
| Λέξεις Κλειδιά: | 6 |
| Abstract..... | 7 |
| Key Words: | 7 |
| 1 BLUETOOTH..... | 11 |
| 1.1 Ιστορική Αναφορά | 11 |
| 1.2 Τι είναι το Bluetooth;..... | 11 |
| 1.3 Σχετικές τεχνολογίες | 12 |
| 1.4 Ζώνη συχνοτήτων | 15 |
| 1.5 Καταναλισκόμενη Ισχύς – Εμβέλεια | 15 |
| 1.6 Bluetooth και ασφάλεια | 15 |
| 1.7 Ακτινοβολία | 18 |
| 1.8 Συσκευές Bluetooth | 21 |
| 1.9 Πιθανές Μελλοντικές Χρήσεις του Bluetooth..... | 24 |
| 1.10 Πλεονεκτήματα Bluetooth | 26 |
| 1.11 Μειονεκτήματα Bluetooth | 27 |
| 2 APXITEKTONIKH BLUETOOTH | 28 |
| 2.1 Εισαγωγή | 28 |
| 2.2 Bluetooth radio..... | 28 |
| 2.3 Χαρακτηριστικά πομπού | 30 |
| 2.3.1 Χαρακτηριστικά διαμόρφωσης..... | 31 |
| 2.3.2 Baseband | 32 |
| 2.3.3 Piconets και Scatternets | 33 |
| 2.3.4 Φυσικά κανάλια και συνδέσεις | 33 |
| 2.3.5 Κωδικός πρόσβασης πακέτων | 35 |
| 2.3.6 Επιγραφή πακέτων | 35 |
| 2.3.7 Packet payload format..... | 37 |
| 2.4 Πρωτόκολλο-link manager (LMP) | 37 |
| 2.4.1 Ασφάλεια | 38 |
| 2.4.2 Τρόποι – Modes | 38 |
| 2.4.3 Σύγχρονες συνδέσεις Connection - Oriented..... | 40 |
| 2.4.4 Στάδια του Bluetooth - Καθιέρωση της σύνδεσης..... | 41 |
| 2.4.5 Άλλα χαρακτηριστικά γνωρίσματα | 43 |
| 2.4.6 Συμπέρασμα..... | 45 |
| 2.5 HOST CONTROLLER INTERFACE..... | 45 |
| 2.5.1 Χαμηλότερα επίπεδα και υλικό | 45 |
| 2.5.2 Εντολές HCI..... | 47 |
| 2.5.3 Επίπεδο μεταφορών (transport layer) HCI USB | 48 |
| 2.5.4 Στρώμα μεταφορών HCI RS- 232 | 49 |
| 2.5.5 Στρώμα μεταφορών HCI UART | 50 |
| 2.5.6 Συμπέρασμα..... | 51 |
| 2.6 Λογικό πρωτόκολλο ελέγχου και προσαρμογής συνδέσεων (L2CAP) | 51 |
| 2.6.1 Λειτουργική περιγραφή | 52 |
| 2.6.2 Κύρια χαρακτηριστικά | 52 |
| 2.6.3 Γεγονότα και ενέργειες | 54 |
| 2.7 RFCOMM..... | 55 |
| 2.7.1 Χαρακτηριστικά γνωρίσματα | 56 |

| | | |
|---------|---|-----|
| 2.7.2 | Υψηλότερα στρώματα | 56 |
| 2.7.3 | Συμπέρασμα..... | 57 |
| 2.8 | Πρωτόκολλο ανακάλυψης υπηρεσιών (Service Discovery Protocol) . | 57 |
| 2.8.1 | Client – server | 58 |
| 2.8.2 | Εγγραφές υπηρεσιών και ιδιότητες υπηρεσιών | 59 |
| 2.8.3 | Κατηγορίες υπηρεσιών (Service Classes)..... | 59 |
| 2.8.4 | Ψάχνοντας και κάνοντας browsing για τις υπηρεσίες | 60 |
| 2.8.5 | Συμπέρασμα..... | 61 |
| 2.9 | Εγκατάσταση Συνδέσεων | 61 |
| 3 | BLUETOOTH PROFILES | 64 |
| 3.1 | Γενικά προφίλ | 65 |
| 3.2 | Προφίλ μεταφορών (Transport Profiles) | 66 |
| 3.2.1 | Προφίλ ώθησης αντικειμένου..... | 67 |
| 3.2.2 | Προφίλ Μεταφοράς Αρχείων..... | 67 |
| 3.2.3 | Προφίλ Συγχρονισμού | 68 |
| 3.3 | Προφίλ τηλεφωνίας (Telephony Profiles) | 68 |
| 3.4 | Προφίλ δικτύωσης | 69 |
| 3.5 | Συμπέρασμα..... | 69 |
| 4 | Τι είναι τα LINUX; | 70 |
| 4.1 | Εισαγωγή | 70 |
| 4.2 | Ιστορία του Linux | 70 |
| 4.2.1 | UNIX..... | 70 |
| 4.2.2 | Linus and Linux | 72 |
| 4.3 | Τι είναι τα LINUX; | 75 |
| 4.3.1 | Τι εννοούμε λέγοντας Open Source λειτουργικό σύστημα; | 76 |
| 4.4 | Τα πλεονεκτήματα του LINUX; | 77 |
| 4.5 | Τα μειονεκτήματα του LINUX;..... | 78 |
| 4.6 | Software και εντολές που υποστηρίζουν τα LINUX | 80 |
| 4.6.1 | Software | 80 |
| 4.6.2 | Εντολές (Commands)..... | 82 |
| 5 | Bluetooth stacks για LINUX..... | 87 |
| 5.1 | OpenBT..... | 87 |
| 5.2 | BlueZ..... | 87 |
| 5.3 | BlueDrekar | 88 |
| 6 | Affix..... | 89 |
| 6.1 | Εισαγωγή | 89 |
| 6.2 | Εισαγωγή στο Affix | 89 |
| 6.2.1 | Αρχιτεκτονική του πυρήνα των Linux..... | 89 |
| 6.2.2 | Affix αρχιτεκτονική | 92 |
| 6.2.3 | Bluetooth and Affix μοντέλα ασφαλείας..... | 94 |
| 6.3 | Εφαρμογές Προγραμματισμού διεπαφών | 96 |
| 6.3.1 | Affix API συστατικά..... | 96 |
| 6.3.2 | Υποδοχή διεπαφής (Socket interface)..... | 98 |
| 6.3.2.1 | Δομές δεδομένων | 99 |
| 6.3.2.2 | Ομάδα συναρτήσεων | 100 |
| 6.3.2.3 | Δείγμα κώδικα | 101 |
| 6.3.2.4 | Επίπεδο ασφαλείας στις υποδοχές..... | 104 |
| 6.3.3 | HCI API | 105 |
| 6.3.3.1 | Ομάδα συναρτήσεων | 105 |
| 6.3.3.2 | Δομές δεδομένων | 110 |

| | | |
|-----------|--|-----|
| 6.3.3.3 | Παράδειγμα..... | 112 |
| 6.3.4 | SDP API..... | 115 |
| 6.3.4.1 | Χαρακτηριστικά γνωρίσματα των υπηρεσιών..... | 115 |
| 6.3.4.2 | SDP δομές δεδομένων και γενικό API | 117 |
| 6.3.4.2.1 | UUID..... | 117 |
| 6.3.4.2.2 | Καταγραφή υπηρεσιών | 118 |
| 6.3.4.2.3 | Δημιουργώντας χαρακτηριστικά γνωρίσματα | 118 |
| 6.3.4.3 | Υπηρεσία πελάτη SDP API | 119 |
| 6.3.4.3.1 | SDP client initialization | 121 |
| 6.3.4.3.2 | Σύνδεση και αποσύνδεση με τον SDP server | 121 |
| 6.3.4.3.3 | Service class identifier(s) based search..... | 122 |
| 6.3.4.3.4 | Αίτηση υπηρεσίας χαρακτηριστικών από ένα συγκεκριμένο αρχείο καταγραφής. | 122 |
| 6.3.4.3.5 | Υπηρεσία class identifier(s) και υπηρεσία service attribute(s) based search | 123 |
| 6.3.4.3.6 | SDP client συναρτήσεις χρησιμότητας..... | 124 |
| 6.3.4.4 | Προμηθευτής υπηρεσιών SDP API | 125 |
| 6.3.4.4.1 | Αρχικοποίηση της SDP υποδομής..... | 126 |
| 6.3.4.4.2 | Σύνδεση και αποσύνδεση με τον SDP server | 126 |
| 6.3.4.4.3 | Δημιουργία και τροποποίηση των χαρακτηριστικών γνωρισμάτων υπηρεσιών | 127 |
| 6.3.4.4.4 | Διαγραφή του service record από τον SDP server..... | 127 |
| 7 | Εκτελεστική Διαδικασία | 128 |
| 7.1 | Σκοπός..... | 128 |
| 7.2 | Εισαγωγή | 129 |
| 7.3 | Προϋποθέσεις και οργάνωση πυρήνων | 130 |
| 7.3.1 | Εγκατάσταση και Μεταγλώττιση | 131 |
| 7.3.2 | Εγκατάσταση απαραίτητων πακέτων | 131 |
| 7.3.3 | Εγκατάσταση του Affix Kernel | 140 |
| 7.3.4 | Εγκατάσταση του Affix | 146 |
| 8 | Σύνδεση USB Bluetooth συσκευής | 162 |
| 8.1 | Εντολές | 162 |
| 8.1.1 | Αναζήτηση γειτονικών συσκευών Bluetooth | 162 |
| 8.1.2 | Εντοπισμός διαθέσιμων υπηρεσιών σε μια εντοπισμένη Bluetooth συσκευή.. .. | 163 |
| 8.1.3 | Σύνδεση με μια Bluetooth συσκευή..... | 164 |
| 8.1.4 | Θέτοντας μία PPP Connection..... | 165 |
| 8.1.5 | Χρησιμοποιώντας το OBEX για μεταφορά αρχείων από και προς το κινητό. | 165 |
| 9 | Ρυθμίσεις κινητού(Τελικό Στάδιο) | 166 |
| 9.1 | GNUBOX | 166 |
| 9.2 | Ρυθμίσεις από την πλευρά του υπολογιστή | 169 |
| 9.3 | Συμπεράσματα | 171 |
| | Βιβλιογραφία | 172 |

1 BLUETOOTH

1.1 Ιστορική Αναφορά

Για εκείνους που ξέρουν λίγα για την τεχνολογία, και ακόμη και για εκείνους που εξοικειώνονται ακόμα περισσότερο με αυτήν, το όνομα Bluetooth ίσως φαίνεται λίγο περίεργο. Το όνομα είναι μια ρομαντική χειρονομία που υπό κάποια έννοια δείχνει τον ενθουσιασμό που η τεχνολογία παράγει τόσο καλά όσο την αξία της ως επαναστατική έννοια. Για να γίνουμε πιο συγκεκριμένη το όνομα το πήρε από τον *Harald I Bluetooth* που ήταν ο βασιλιάς της Δανίας από το 940 έως 985 μ.Χ. Ο *Harald I Bluetooth* ένωσε τη Δανία και τη Νορβηγία. Το Bluetooth σήμερα προσδοκεί στο να ενώσει τους κόσμους των υπολογιστών και των τηλεπικοινωνιών (ενδεχομένως περισσότερο από το λίγο που το βασίλειο Harald's Viking άκμασε.) Το 1994 η κινητή επικοινωνία Ericsson άρχισε μια μελέτη για να ερευνηθεί η δυνατότητα πραγματοποίησης μιας χαμηλής ισχύος και χαμηλού κόστους ράδιο-ζεύξης μεταξύ των κινητών τηλεφώνων και των εξαρτημάτων τους. Τον Φεβρουάριο του 1998, πέντε επιχειρήσεις, η Ericsson, η IBM (ΗΠΑ), η Intel (ΗΠΑ), η Nokia (Φιλανδία) και η Toshiba (Ιαπωνία) διαμόρφωσαν μια ειδική ομάδα ενδιαφέροντος (SIG). Η ομάδα περιείχε τα απαραίτητα μέλη επιχειρησιακού τομέα - δύο πρωτοπόρους στην αγορά της κινητή τηλεφωνίας, δύο πρωτοπόρους στην αγορά των υπολογιστών (φορητών και μη) και έναν πρωτοπόρο στην αγορά της επεξεργασίας ψηφιακού σήματος. Σκοπός ήταν η ενσωμάτωση του Bluetooth σε μεγάλο αριθμό διαφορετικών συσκευών και η συμβατότητα μεταξύ αυτών.

1.2 Τι είναι το Bluetooth;

Το Bluetooth είναι μια ανοικτή τυποποιημένη προδιαγραφή μιας ραδιοσυχνότητας (RF) περιορισμένου φάσματος. Υπόσχεται να αλλάξει το πρόσωπο της ασύρματης επικοινωνίας και των υπολογιστών. Είναι ένα ανέξοδο, ασύρματο σύστημα δικτύωσης για όλες τις κατηγορίες φορητών συσκευών, όπως τα Lap-Tops, τα PDAs (προσωπικοί ψηφιακοί βοηθοί), και τα κινητά τηλέφωνα που βοηθάει στη σύνδεση και στη μετάδοση στοιχείων μεταξύ αυτών των συσκευών. Έχει ως σκοπό την επίτευξη ασύρματων συνδέσεων για τους

υπολογιστές γραφείου, κάνοντας τις συνδέσεις μεταξύ των οργάνων ελέγχου, τους εκτυπωτές, τα πληκτρολόγια, και την ΚΜΕ χωρίς τη χρήση καλωδίων και ό,τι αυτό συνεπάγεται.

Αυτή η ιδέα της ασύρματης τεχνολογίας συλλήφθηκε αρχικά από την Ericsson το 1994, όταν άρχισε η ίδια η επιχείρηση μια μελέτη για να ερευνήσει τη δυνατότητα πραγματοποίησης μιας χαμηλής ισχύος και χαμηλού κόστους ράδιο-επαφής μεταξύ των κινητών τηλεφώνων και των εξαρτημάτων τους. Ο στόχος της εταιρίας ήταν να εξαλειφτεί η ανάγκη για τα καλώδια. Το προβαλλόμενο κόστος του τσιπ ήταν περίπου \$5, και επρόκειτο να απαιτήσει τη χαμηλότερη δυνατή δύναμη έτσι ώστε να μπορεί να χρησιμοποιηθεί στις συσκευές που στηρίζονται στη ζωή μπαταριών.

1.3 Σχετικές τεχνολογίες

➤ IrDA

Τα αρχικά IrDA προέρχονται από το Infrared Data Association. Το IrDA επιτρέπει την αμφίδρομη υπέρυθρη επικοινωνία μεταξύ των συσκευών. Η ταχύτητες που επιτυγχάνουμε με τις υπέρυθρες ακτίνες κυμαίνονται από: 9600 bps έως 4 Mbps. Όπως και το Bluetooth έτσι κι αυτό έχει κάνει αισθητή την παρουσία του στην αγορά, από τα κινητά τηλέφωνα μέχρι και τους υπολογιστές, όμως όλα δείχνουν πως δεν υπάρχει μέλλον σε αυτό τον τρόπο σύνδεσης. Το βασικό μειονέκτημα των υπέρυθρων είναι η μικρή εμβέλεια, η οποία κυμαίνεται στα ένα με δύο μέτρα και στο ότι πρέπει να υπάρχει σύνδεση στην οποία να μην παρεμβάλλονται εμπόδια, σε αντίθεση με το Bluetooth το οποίο έχει εμβέλεια από 10 μέχρι 250 μέτρα και μπορεί να γίνει σύνδεση μεταξύ συσκευών και ας παρεμβάλλονται εμπόδια. Το κύριο πλεονέκτημα που έχει και ίσως το μόνο που το κρατάει ακόμα «ζωντανό» είναι η υψηλή ταχύτητα που επιτυγχάνει και φτάνει να είναι και τέσσερις φορές μεγαλύτερη από αυτή που έχουν αντίστοιχες τεχνολογίες.

➤ **HomeRF**

To HomeRF Shared Wireless Access Protocol (SWAP) σύστημα σχεδιάστηκε για να μεταφέρει και φωνή και δεδομένα και να μπορεί να επικοινωνήσει με το δημόσιο δίκτυο τηλεφώνου Public Switched Telephone Network (PSTN) και κατ' επέκταση με το Internet. Όπως και το Bluetooth έτσι κι αυτό λειτουργεί στη ζώνη των 2,4 GHz. Η τεχνολογία SWAP απορρέει από την επέκταση του υπάρχοντος ασύρματου τηλεφώνου (Digital Enhanced Cordless Telephone or DECT) και από την ασύρματη τεχνολογία δικτύου (Wireless LAN) για να καθιστά ικανή μια νέα κατηγορία εγχώριων ασύρματων υπηρεσιών. Το SWAP υποστηρίζει και TDMA (Time Division Multiple Access) υπηρεσίες για τη μετάδοση φωνής και άλλων παρεμφερή υπηρεσιών και CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance) υπηρεσίες για τη γρήγορη μετάδοση δεδομένων.

Η κατανάλωση ισχύος αυτής της συσκευής είναι γύρω στα 100mW και οι ταχύτητες μετάδοσης δεδομένων κυμαίνονται από 1Mbps έως 2 Mbps ανάλογα με τις τεχνικές διαμόρφωσης που χρησιμοποιούνται. Η χαρακτηριστική χρήση αυτής της τεχνολογίας είναι μέσα στο σπίτι, και το δίκτυο μπορεί να υποστηρίξει μέχρι 127 συσκευές. Το δίκτυο HomeRF μπορεί να υποστηρίξει μέχρι 6 πλήρως διπλά κανάλια φωνής. Αναλυτικά τα χαρακτηριστικά των τεχνολογιών αυτών φαίνονται στο πίνακα I.

| Specification | IrDA | Bluetooth | HomeRF |
|--|-----------|-----------|--------|
| Data Rate (Kbps) | 4000 | 1000 | 2000 |
| Distance (m) | 1 | 10 | 50 |
| Approximate Costs of transceiver and controller (US\$) | 3.5 – 5.5 | 20.00 | 25.00 |

| | | | |
|---------------------|----------------|---------------------|------------|
| Number of Devices | 2 | 8 | 127 |
| Isochronous traffic | Yes | Yes | Yes |
| Voice Channels | 1 | 3 | 6 |
| Through Walls | No | Yes | Yes |
| Spectrum | Optical 850 nm | RF 2.4 GHz | RF 2.4 GHz |
| Topology | Point-to-Point | Point-to-multipoint | Network |

ΠΙΝΑΚΑΣ Ι

Στο μέλλον, το Bluetooth είναι πιθανό να είναι το πρότυπο στις δεκάδες των εκατομμυρίων των κινητών τηλεφώνων, τα PCs, τα Lap-Tops και μιας σειράς άλλων ηλεκτρονικών συσκευών. Η τεχνολογία Bluetooth είναι πιθανό να συνυπάρξει με άλλες ασύρματες τεχνολογίες, όπως IrDA και HomeRF. Η IrDA, η οποία είναι και αυτή εγκατεστημένη σε εκατομμύρια συσκευές όπως το Bluetooth, έχει τον περιορισμό της επικοινωνίας οπτικής επαφής, ο οποίος δεν είναι έτσι στην περίπτωση των Bluetooth και HomeRF. Αφ' ετέρου, λόγω στην ιδιαίτερα κατευθυντική φύση της μετάδοσης (της ακτίνας IR), IrDA είναι λιγότερη επιρρεπής σε παρέμβαση από τις παρόμοιες συσκευές στις γειτονικές περιοχές και πετυχαίνει και μεγάλες ταχύτητες. Σε συνδυασμό με άλλες τεχνολογίες, όπως WAP (ασύρματο πρωτόκολλο εφαρμογής) και Symbian, το Bluetooth θα έχει τεράστια αποτελέσματα στην καθημερινή ζωή. Το Bluetooth είναι μια από τις βασικές τεχνολογίες που μπορούν να καταστήσουν την κινητή κοινωνία των πληροφοριών πιθανή, θολώνοντας τα όρια μεταξύ του σπιτιού, του γραφείου, και του εξωτερικού κόσμου.

1.4 Ζώνη συχνοτήτων

Οι συσκευές που χρησιμοποιούν Bluetooth εκπέμπουν και λαμβάνουν σήματα στη περιοχή των 2.4 GHz η οποία αποτελεί μια ελεύθερη Industrial Scientific and Medical (ISM) ζώνη συχνοτήτων. Στη ζώνη αυτή έχουμε κάποιους κανόνες όσον αφορά τη ισχύ και το εύρος του φάσματος του μεταδιδόμενου σήματος για να αποφεύγονται οι παρεμβολές.

1.5 Καταναλισκόμενη Ισχύς – Εμβέλεια

Το βασικό χαρακτηριστικό του Bluetooth είναι ότι καταναλώνει χαμηλή ενέργεια, αφού χρησιμοποιείται σε συσκευές όπως κινητά τηλέφωνα PDA's και άλλες παρόμοιες συσκευές που η κατανάλωση ενέργειας είναι σημαντικός παράγοντας. Για να γίνει αυτό δυνατό, οι προδιαγραφές καθορίζουν 3 διαφορετικούς τρόπους λειτουργίας όσον αφορά τη κατανάλωση ενέργειας :

- *1mW (0 dBm) που είναι η βασική στάθμη*
- *2.5mW (+4 dBm)*
- *100mW (+20 dBm)*

Οι παραπάνω καταστάσεις λειτουργίας παρέχουν στις συσκευές ακτίνα λειτουργίας που ξεκινάει από τα 10 με 20 μέτρα και φτάνει τα 250 μέτρα.

1.6 Bluetooth και ασφάλεια

Στη σημερινή κοινωνία ο όρος ηλεκτρονικό εμπόριο χρησιμοποιείται όλο και περισσότερο. Στο μέλλον δεν είναι απίθανο να χρησιμοποιούμε προσωπικές ηλεκτρονικές συσκευές για να πληρώνουμε λογαριασμούς, αντί για χρήματα, πιστωτικές κάρτες κ.λ.π. Σήμερα πολλοί χρησιμοποιούν το Internet για να πληρώσουν τους λογαριασμούς αυτό σημαίνει ότι πρέπει να έχουμε εμπιστοσύνη στις ηλεκτρονικές συσκευές. Έτσι, η ασφάλεια ή ζητήματα ασφάλειας στα δίκτυα (Διαδίκτυο, κινητά δίκτυα κ.λ.π.) έχει γίνει πολύ σημαντική. Πολλές επιχειρήσεις

που αναπτύσσουν εφαρμογές Bluetooth θα πρέπει να εξετάσουν λεπτομερώς όλες τις πτυχές ασφάλειας.

Είναι φανερό ότι η ασφάλεια που παρέχεται από τα σταθερά καλώδια στα συνδεδεμένα με καλώδιο δίκτυα πρέπει να αντικατασταθεί με άλλα μέσα ασφάλειας στο ασύρματο περιβάλλον. Ένα ράδιο σήμα μπορεί εύκολα να παρεμποδιστεί ή να κλαπεί, επομένως είναι μεγάλης σπουδαιότητας οι Bluetooth συσκευές να έχουν μεθόδους για να αποτρέψουν την υποκλοπή, την τροποποίηση των στοιχείων ή τη χρήση ψεύτικης ταυτότητας. Θα πρέπει να υπάρχουν λοιπόν μηχανισμοί ασφάλειας για να παρέχουν πιστοποίηση του χρήστη και κρυπτογράφηση των δεδομένων. Όλες οι συσκευές θα πρέπει να υποστηρίζουν τους μηχανισμούς ασφάλειας Bluetooth.

Η ασφάλεια στις συσκευές Bluetooth υπάρχει σε διάφορα επίπεδα, αλλά η ασφάλεια Bluetooth δεν προορίζεται να αντικαταστήσει τα υπάρχοντα χαρακτηριστικά γνωρίσματα ασφάλειας δικτύων. Επιπρόσθετη ασφάλεια επιπέδου εφαρμογής μπορεί να εφαρμοστεί για πρόσθετη υψηλότερη ασφάλεια, όπως π.χ για το ηλεκτρονικό εμπόριο. Οι κυριότερες μέθοδοι για παροχή ασφάλειας σε συσκευές Bluetooth είναι:

- Κάθε μονάδα έχει μια μοναδική διεύθυνση Bluetooth (BD_ADDR).
- Κώδικες ΑΣΦΑΛΕΙΑΣ (προσωπικός αριθμός ταυτότητας - PIN) για set-up διαδικασίες.
- Κλειδιά για την επικύρωση και την κρυπτογράφηση.
- Προκαθορισμένοι αλγόριθμοι για την επικύρωση και την κρυπτογράφηση.
- Εφαρμογή πρόσθετης ασφάλειας επιπέδου εφαρμογής.
- Εναλλαγή συχνότητας (frequency hopping).
- Περιορισμοί της εμβέλειας μετάδοσης.

Δεδομένου ότι οι διάφορες εφαρμογές έχουν διαφορετικές απαιτήσεις στην ασφάλεια, απαιτείται ευελιξία στη ασφάλεια του link επιπέδου.

Με βάση αυτή την λογική καθορίζονται τρία επίπεδα ασφάλειας, που αναφέρονται ως **Bluetooth Security Modes**:

1) **Χωρίς ασφάλεια:** Μια συσκευή που έχει αυτό το mode ασφάλειας δεν θα κινήσει οποιαδήποτε διαδικασία ασφάλειας. Αυτός ο τρόπος ασφάλειας χρησιμοποιείται συνήθως κατά την εξέταση των μη ευαίσθητων πληροφοριών. Παρακάμπτει τις λειτουργίες ασφάλειας του link-επίπεδου έτσι ώστε καμία διαδικασία ασφάλειας να μην αρχικοποιείται. Ο τρόπος ασφάλειας 1 μπορεί να θεωρηθεί ως ειδική περίπτωση του τρόπου ασφάλειας 2 όπου καμία υπηρεσία δεν απαιτεί οποιαδήποτε ασφάλεια.

2) **Ασφάλεια που επιβάλλεται από το service-level:** Μια συσκευή αυτού του mode ασφάλειας δεν αρχίζει διαδικασίες ασφάλειας πριν από την καθιέρωση καναλιών σε επίπεδο L2CAP. Εάν μια διαδικασία ασφάλειας εκκινείται ή όχι εξαρτάται από τις απαιτήσεις ασφαλείας του καναλιού ή της υπηρεσίας. Αυτός ο τρόπος ασφάλειας χρησιμοποιείται κυρίως όταν εφαρμογές με διαφορετικές απαιτήσεις ασφάλειας τρέχουν παράλληλα

3) **Ασφάλεια που επιβάλλεται από το link-level:** Μια συσκευή αυτού του mode ασφάλειας κινεί τις διαδικασίες ασφάλειας πριν ολοκληρωθεί η οργάνωση συνδέσεων σε επίπεδο LMP. Με αυτόν τον τρόπο ο link manager επιβάλλει την ασφάλεια σε κοινό επίπεδο για όλες τις εφαρμογές στην αρχή της σύνδεσης. Ο τρόπος αυτός είναι λιγότερο εύκαμπτος από τον τρόπο 2 αλλά ευκολότερος να εφαρμοστεί και καταλληλότερος για ένα κοινό επίπεδο ασφάλειας. Ένα σύστημα που αναπτύσσει δραστηριότητες χρησιμοποιώντας αυτόν τον τρόπο ασφάλειας, μπορεί είτε να ζητήσει μόνο πιστοποίηση, είτε και πιστοποίηση και κρυπτογράφηση των δεδομένων.

Ανεξάρτητα από το mode ασφάλειας που χρησιμοποιείται, υπάρχει πάντα η πιθανότητα ο χρήστης μιας συσκευής Bluetooth να θέσει σε κίνδυνο την ασφάλειά της. Το δυσάρεστο μάλιστα είναι ότι η αποφυγή αυτού του γεγονότος

είναι σχεδόν αδύνατη. Όπως σε όλες άλλωστε τις υπηρεσίες που χρειάζεται ασφάλεια. Παραδείγματος χάριν μπορεί να υπάρξουν προβλήματα σχετικά με την ασφάλεια, εάν ο χρήστης μιας συσκευής νομίζει ότι χρησιμοποιεί μια ασφαλή υπηρεσία, αλλά στην πραγματικότητα δεν είναι ασφαλής, εξαιτίας κάποιου λάθους που έχει κάνει. Επίσης υπάρχει πάντα το πρόβλημα να υποκλαπεί ο κωδικός PIN. Ανεξάρτητα από το εάν η κακή χρήση ήταν μια πράξη άγνοιας ή όχι, αυτά είναι εμπόδια που οι υπεύθυνοι για την ανάπτυξη πρέπει να λάβουν υπόψιν τους. Υπάρχουν πάντα τρόποι για να γίνει το λάθος αλλά το ζητούμενο είναι να καταβληθούν προσπάθειες ώστε να ελαχιστοποιηθεί η ζημία που προκαλείται από αυτό.

1.7 Ακτινοβολία

Τον τελευταίο καιρό, με την ραγδαία ανάπτυξη της ασύρματης επικοινωνίας και της κινητής τηλεφωνίας, δημιουργείται πολύ θόρυβος για το πόσο η ακτινοβολία βλάπτει την ανθρώπινη υγεία. Έχουν ακουστεί πολλά, άλλα υπερβολικά και άλλα τρομακτικά που έχουν όμως μια βάση. Η ασύρματη επικοινωνία, έχει μπει στη ζωή μας τόσο πολύ τις τελευταίες δεκαετίες με τη χρήση των κινητών τηλεφώνων. Οπότε είναι δύσκολο έως αδύνατο να αποφανθούμε αν κάνουν κακό στην υγεία του ανθρώπου, αν ναι πόσο και πιο είναι το όριο που μπορούμε να πούμε πως νοιώθουμε ασφαλείς. Σήμερα υπάρχουν κάποια όρια, αλλά χωρίς να ξέρουμε με βεβαιότητα αν και τηρώντας αυτά τα όρια, μπορούμε να αισθανόμαστε ασφαλείς.

Η περισσότερη κριτική έχει σαν στόχο τα κινητά τηλέφωνα, πολλά όμως άρθρα ειδήσεων και πολλές έρευνες ασχολούνται ήδη με τα ζητήματα προσωπικής ασφάλειας που προκύπτουν από την χρήση της τεχνολογίας Bluetooth. Η βασική ερώτηση που προκύπτει από αυτές τις κριτικές είναι αν το Bluetooth προκαλεί οποιουδήποτε είδους επιπτώσεις στην υγεία. Πολλές συσκευές Bluetooth χρησιμοποιούνται και φοριούνται κοντά στο σώμα. Παραδείγματος χάριν στο Ericsson Bluetooth headset, το τσιπ Bluetooth βρίσκεται πολύ κοντά στο αυτί. Το Bluetooth χρησιμοποιεί τις ίδιες συχνότητες με τους φούρνους μικροκυμάτων και οι φούρνοι μικροκυμάτων χρησιμοποιούνται στη θέρμανση των τροφίμων σε πολύ υψηλές θερμοκρασίες.

Μπορεί αυτό να είναι πραγματικά καλό; Η εκλυόμενη ισχύς ενός φούρνου μικροκυμάτων είναι περίπου 1000 W, τη στιγμή μια συσκευή Bluetooth για κανονικές εφαρμογές χρησιμοποιεί μόνο 1 mW (0 dBm). Η διαφορά είναι τεράστια και έτσι δεν πρέπει να ανησυχήσουμε τουλάχιστον ότι υπάρχει περίπτωση να ψηθούμε.

Η σύγκριση των συμβατικών κινητών τηλεφώνων με τα προϊόντα Bluetooth είναι επίσης ενδιαφέρουσα. Η μέγιστη ισχύς παραγωγής για ένα κινητό τηλέφωνο που χρησιμοποιεί τις συχνότητες από 450 έως 2200 MHz είναι μεταξύ 10 mW και 2 W. Η εκλυόμενη ισχύς για μια συσκευή Bluetooth είναι κανονικά 1 mW (0 dBm) και για την εκτεταμένη σειρά (εμβέλεια 100m) μόνο 100 mW (20 dBm). Και σε αυτήν την περίπτωση μπορούμε να δούμε ότι η διαφορά μεταξύ 100 mW και 2 W είναι αρκετά μεγάλη.

Μια άλλη σημαντική ερώτηση είναι εάν η εκπομπή ραδιοσυχνότητας (RF) θερμαίνει το ανθρώπινο σώμα. Γνωρίζουμε όμως ότι το σώμα απορροφά μόνο ένα μέρος της εκλυόμενης ισχύος. Για ένα φορητό κινητό τηλέφωνο η θερμοκρασία του σώματος αυξάνεται κατά 0.1 βαθμούς Κελσίου και για το Bluetooth δεν υπάρχει καμία ανιχνεύσιμη αύξηση θερμοκρασίας. Αυτό δεν δηλώνεται μόνο από τους κατασκευαστές προϊόντων Bluetooth, όπως την Ericsson, αλλά και από τα διεθνή πρότυπα και τις συστάσεις έκθεσης RF που έχουν αναπτυχθεί από την Παγκόσμια Οργάνωση Υγείας (WHO), την Ευρωπαϊκή Επιτροπή και το IEEE. Για να μετρηθεί η απορρόφηση RF χρησιμοποιείται ο όρος Specific Absorption Rate (SAR). Το SAR εκφράζει με σύντομους όρους το ποσοστό στο οποίο εμφανίζεται η αύξηση της θερμοκρασίας του σώματος. Είναι αδύνατο για τις ραδιο συσκευές με ισχύ μικρότερη από 1.6 mW να υπερβούν τα όρια SAR. Αυτό έχει σαν αποτέλεσμα εκείνες οι συσκευές με εκλυόμενη ισχύ κάτω από 1.6 mW (όπως μια κανονική εφαρμογή Bluetooth, 1 mW) να μην χρειάζονται οποιαδήποτε δοκιμή! Οι συσκευές Bluetooth της εκτεταμένης σειράς (εκλυόμενης ισχύος 100 mW) πρέπει να υποβληθούν στα πρότυπα και τις οδηγίες έκθεσης. Για μερικές από αυτές τις συσκευές η συμβατότητα πρέπει να επιτευχθεί με τον κατάλληλο σχεδιασμό και δοκιμή. Οι δοκιμές που εκτελούνται π.χ από την Ericsson στο Ultimate Headset με εκλυόμενη ισχύ 1 mW, δείχνουν ότι η τιμή SAR είναι 1/100th, μικρότερη δηλαδή από το αποδεκτό όριο.

Η έκθεση RF από τις συσκευές Bluetooth φαίνεται να μην δημιουργεί κανένα πρόβλημα. Από την άλλη όμως υπάρχουν ορισμένοι τομείς οι οποίοι χρειάζονται περαιτέρω αξιολόγηση όσον αφορά τις επιπτώσεις στην υγεία, όπως παραδείγματος χάριν η παρεμβολή στους βηματοδότες και στους ενισχυτές ακρόασης ή η παρεμβολή στις συσκευές νοσοκομείων. Σε αυτούς τους τομείς απαιτείται περισσότερη έρευνα, καλή χρήση της εφαρμοσμένης μηχανικής και σωστός συντονισμός της βιομηχανίας.

Σε γενικές πάντως γραμμές οι ακόλουθοι τρεις παράγοντες αποδεικνύουν ότι το Bluetooth είναι ασφαλές:

- 1) Τα επίπεδα έκθεσης σε ακτινοβολία, εξαιτίας των συσκευών Bluetooth είναι κατά πολύ κάτω από τα υπάρχοντα όρια ασφάλειας.
- 2) Καμία δοκιμή δεν απαιτείται στο επίπεδο παραγωγής για τις τυποποιημένες συσκευές Bluetooth.
- 3) Διάφορες οργανώσεις υγείας υποστηρίζουν τα πρότυπα και τις συστάσεις συσκευών RF Bluetooth αν και ενάντια τους έχουν καταφερθεί πολλοί φορείς.

Πολλοί διαφορετικοί φορείς, ο καθένας από την δική του οπτική γωνία (υπεύθυνοι για την ανάπτυξη, κατασκευαστές και διαφορετικές οργανώσεις υγείας) έχουν καταλήξει στο ίδιο συμπέρασμα: Το Bluetooth δεν είναι στην παρούσα μορφή του ένας κίνδυνος για την ανθρώπινη υγεία. Το επόμενο βήμα που οφείλουν να κάνουν είναι να εκπαιδεύσουν και να πείσουν τους καταναλωτές γι' αυτό.

1.8 Συσκευές Bluetooth

Κινητά τηλέφωνα Bluetooth

Πολλά κινητά τηλέφωνα έχουν ενσωματωμένο Bluetooth. Αρχικά, αρκούσε η προσθήκη ενός Bluetooth adapter στο κινητό, που του έδινε την δυνατότητα της Bluetooth επικοινωνίας. Η Cambridge Silicon Radio όμως, είχε την ιδέα το Bluetooth hardware να ενσωματώνεται στην μπαταρία του τηλεφώνου, ώστε το τελευταίο να έχει build-in Bluetooth υποστήριξη.



Personal Digital Assistants (PDAs)

Μια όλο και περισσότερο δημοφιλής φορητή συσκευή είναι το PDA. Στην σημερινή αγορά υπάρχουν αρκετοί κατασκευαστές PDAs που ανταγωνίζονται. Για να προελκύσουν τους καταναλωτές πρέπει να υποστηρίζουν την τεχνολογία Bluetooth. Ένα PDA χρησιμοποιείται ως ημερολόγιο, σημειωματάριο, ως συσκευή για να γράψει κανείς τα mail του κ.λπ. Με την ενσωμάτωση του Bluetooth μπορεί να χρησιμοποιηθεί και με πολλούς άλλους τρόπους, παραδείγματος χάριν για να συγχρονιστεί με άλλες συσκευές ή για να στείλει τα mail αμέσως. Το PDA μπορεί να έχει Bluetooth υποστήριξη είτε με ένα plug-on module, είτε έχοντας ενσωματωμένο το bluetooth module. Η εταιρία Palm Inc. είναι ένας κατασκευαστής που αναπτύσσει ένα Bluetooth PDA.



PC-card

Μια PC-κάρτα Bluetooth, αποκαλούμενη επίσης PCMCIA-κάρτα, χρησιμοποιείται στα laptop ή τα notebook PCs για διαφορετικές ασύρματες υπηρεσίες. Η PC-κάρτα είναι ένας απλός τρόπος που επιτρέπει σε ένα notebook να έχει Bluetooth που μπορεί να χρησιμοποιηθεί σε διάφορα πεδία εφαρμογών. Μερικές από αυτές είναι η πρόσβαση στο Internet, μέσω Network Access Points ή ενός κινητού τηλεφώνου (WAP), η πρόσβαση στους εκτυπωτές, ο συγχρονισμός μεταξύ συσκευών και η λήψη του ηλεκτρονικού ταχυδρομείου. Πολλές εταιρίες κατασκευάζουν PC-κάρτες με παρόμοια λειτουργικότητα.



Φορητή συσκευή αποθήκευσης

Η σουηδική επιχείρηση Tactel έχει αναπτύξει μια φορητή συσκευή αποθήκευσης για την Ericsson, με σκοπό να αναδείξει την τεχνολογία Bluetooth. Η συσκευή έχει αποθηκευτικό χώρο 192 MB και ονομάζεται NetDrive. Το NetDrive έχει επίσης ενσωματωμένο έναν WEB Server και υποστηρίζει υπηρεσίες, FTP και Telnet.



Dongle USB

Για να εξασφαλιστεί η Bluetooth υποστήριξη σε ένα notebook ή Desktop PC, ώστε να είναι δυνατή η ασύρματη επικοινωνία του με διάφορα είδη συσκευών (απομακρυσμένες συσκευές, modems, κινητά τηλέφωνα κ.λπ.), απαιτείται ένας προσαρμοστής Bluetooth που συνδέεται στο PC. Ένας προσαρμοστής κατάλληλος για αυτόν τον σκοπό είναι το dongle USB. Διάφοροι κατασκευαστές παράγουν dongles USB, μεταξύ των οποίων είναι: Motorola, DigiAnswer, Psion Dacom, WIDCOMM, INC, κ.τ.λ.



RS232 dongle

Πρόκειται για μια συσκευή σειριακής επικοινωνίας, η οποία λειτουργεί σχεδόν παρόμοια με το USB dongle. Το dongle RS232 είναι μια συσκευή που συνδέεται απλά με τη σειριακή θύρα π.χ ενός PC. Κυρίως χρησιμοποιείται σε εφαρμογές που προσομοιώνουν ένα καλώδιο, όπως π.χ αυτό ενός Bluetooth modem.



1.9 Πιθανές Μελλοντικές Χρήσεις του Bluetooth

Το Bluetooth κάνει την ζωή ευκολότερη ...

Στο γραφείο...

Φθάνετε στο γραφείο και τοποθετείτε τον χαρτοφύλακά σας στο γραφείο σας. Αυτόματα, ο προσωπικός ψηφιακός βοηθός σας (PDA) συγχρονίζεται με τον προσωπικό υπολογιστή σας και μεταφέρει σε αυτόν (με την βοήθεια της τεχνολογίας Bluetooth) τα αρχεία, το ηλεκτρονικό ταχυδρομείο και το πρόγραμμα του ημερολογίου.



Κατά τη διάρκεια μιας συνεδρίασης, έχετε πρόσβαση στο PDA σας για να στείλετε την παρουσίασή σας στο ηλεκτρονικό whiteboard. Καταγράφετε τα πρακτικά της συνεδρίασης στο PDA σας και τα μεταφέρετε ασύρματα στους συμμετέχοντες πριν αποχωρήσουν από τη συνεδρίαση.

Είστε ο επόπτης εργοστασίων για Widgets, INC. Καθώς περπατάτε μέσα στο εργοστάσιο, είστε σε θέση να ελέγξετε την κατάσταση κάθε κομματιού του εξοπλισμού δοκιμής αφού μπορείτε αμέσως να κατεβάσετε ένα user interface για κάθε μηχανή. Μπορείτε ακόμα να ζητήσετε τα ποσοστά ατέλειας των προϊόντων και τις επιμέρους αποτυχίες του κάθε κομματιού στους επιλεγμένους τερματικούς σταθμούς.



Στο σπίτι ...



Κατά την άφιξη στο σπίτι σας, η πόρτα ξεκλειδώνει αυτόματα για σας, τα φώτα εισόδου ανοίγουν αυτόματα, και η θερμοκρασία προσαρμόζεται στις καθορισμένες από εσάς τιμές.

Ένας συναγερμός σας ειδοποιεί ότι το μικρό παιδί σας έφυγε από το σπίτι.



Η χρήση του PDA μετατρέπεται από επιχειρηματική σε προσωπική καθώς μπαίνετε στο σπίτι σας. Ένας ηλεκτρονικός πίνακας δελτίων στο σπίτι προσθέτει αυτόματα τις σχεδιασμένες δραστηριότητές στο οικογενειακό ημερολόγιο, και σας προειδοποιεί για οποιεσδήποτε εκκρεμότητες.

Έχετε ένα σύστημα ασφάλειας για το σπίτι που αποτελείται από συσκευές τεχνολογίας Bluetooth. Μόλις αναβαθμίσετε το σύστημα και προσθέσετε νέες συσκευές. Επειδή χρησιμοποιείται τεχνολογία Bluetooth, το σύστημα αναπροσαρμόζεται αυτόματα και οι συσκευές αναγνωρίζουν η μια την άλλη.



Πολλές από τις παραπάνω χρήσεις ίσως φαίνονται ακραίες άλλες πάλι είναι πιο κοντά στην πραγματικότητα. Το μόνο σίγουρο είναι ότι το Bluetooth ανοίγει νέους δρόμους και ορίζοντες.

1.10 Πλεονεκτήματα Bluetooth

Οι συσκευές που χρησιμοποιούν Bluetooth εκπέμπουν και λαμβάνουν σήματα στην περιοχή των 2.4 GHz η οποία αποτελεί μια ελεύθερη Industrial Scientific and Medical (ISM) ζώνη συχνοτήτων. Η λειτουργία ενός προϊόντος σε αυτήν την ζώνη μπορεί να γίνει χωρίς την αγορά των αδειών ή την πληρωμή των δικαιωμάτων, σε παγκόσμιο επίπεδο, σε αντίθεση με όλες τις υπόλοιπες ζώνες συχνοτήτων. Το Bluetooth είναι μια φιλική προς το χρήστη ράδιο – διεπαφή, η οποία αν και αρχικά σχεδιάστηκε για την αντικατάσταση των καλωδίων στην κινητή τηλεφωνική αγορά, έχει αναβαθμιστεί από τότε σε μια περιορισμένου φάσματος ασύρματη λύση δικτύωσης. Η τεχνική μεταπήδησης συχνότητας που χρησιμοποιεί, το καθιστά πολύ ανθεκτικό ενάντια στην παρεμβολή. Αυτή η ανθεκτικότητα στην παρεμβολή είναι πρώτηστη απαίτηση στη ISM ζώνη εξαιτίας του μεγάλου αριθμού προϊόντων που λειτουργούν σε αυτήν την ζώνη.

Το πρότυπο Bluetooth υποστηρίζει τόσο τη μετάδοση δεδομένων όσο και τη μετάδοση φωνής. Το γεγονός αυτό είναι πολύ σημαντικό και αρκεί για να καταστήσει το Bluetooth τυποποιημένο πρότυπο απέναντι σε άλλα ανταγωνιστικά πρότυπα που υποστηρίζουν μόνο ένα είδος επικοινωνίας.

Τα modules ή τα τσιπς Bluetooth είναι μικρά, ανθεκτικά στην παρέμβολή και καταναλώνουν πολύ λίγη ενέργεια. Αυτό είναι το κύριο χαρακτηριστικό που καθιστά το Bluetooth μια ιδανική λύση για τις φορητές ασύρματες συσκευές όπως οι υπολογιστές χειρός (PDAs), τα headsets, τα κινητά τηλέφωνα κ.λπ. Δεδομένου ότι η εκλύομενη ενέργεια είναι πολύ μικρότερη από αυτή ενός κινητού τηλεφώνου, το Bluetooth δίνει τη δυνατότητα ασύρματης επικοινωνίας χωρίς οποιουδήποτε πιθανούς κινδύνους της ακτινοβολίας RF.

Το Bluetooth είναι ένα ανοικτό πρότυπο. Γι' αυτό είναι ελκυστικό για τους κατασκευαστές και σε συνδυασμό με την αυστηρή διαδικασία προσόντων (qualification process), εξασφαλίζει διαλειτουργικότητα (interoperability) μεταξύ συσκευών διαφορετικών ειδών και διαφορετικών κατασκευαστών. Οι υπηρεσίες που παρέχονται από το Bluetooth είναι ήδη εκτενείς και ο αριθμός τους αναμένεται να αυξηθεί εντυπωσιακά. Ένα άλλο ελκυστικό χαρακτηριστικό γνώρισμα για τους κατασκευαστές είναι ότι στο εγγύς μέλλον προβλέπεται γρήγορη απορρόφηση των προϊόντων που χρησιμοποιούν Bluetooth.

Ένας παράγοντας που πρέπει πάντα να εξετάζεται είναι εάν μια αγορά θα δεχτεί ή όχι τη νέα τεχνολογία. Στην περίπτωση του Bluetooth που 1900 επιχειρήσεις έχουν υπογράψει ήδη μια συμφωνία adopters, και πολλές από αυτές αναπτύσσουν αυτήν την περίοδο τα προϊόντα τους, ενσωματώνοντας Bluetooth τεχνολογία, τα πρώτα σημάδια είναι παραπάνω από ενθαρρυντικά. Ο μεγάλος αριθμός adopters σε ένα τόσο σύντομο χρονικό διάστημα είναι πρωτοφανής στη βιομηχανική ιστορία και είναι μια πολύ ελπιδοφόρος έναρξη για την τεχνολογία Bluetooth.

1.11 Μειονεκτήματα Bluetooth

Τα πρότυπα Bluetooth έχουν μια εμβέλεια περίπου 10 μέτρων. Αν και υπάρχει η δυνατότητα επέκτασης στα 250 μέτρα, η απόσταση αυτή είναι μικρή συγκρινόμενη με λύσεις που προσφέρουν άλλες ασύρματες τεχνολογίες, με εξαίρεση τις υπέρυθρες ακτίνες (infrared). Επιπλέον τα εμπόδια, όπως οι τοίχοι, μπορούν να μειώσουν την εμβέλεια.

Για τις φορητές συσκευές υπάρχει πάντα το ζήτημα του χρόνου της επαναφόρτισης των μπαταριών.

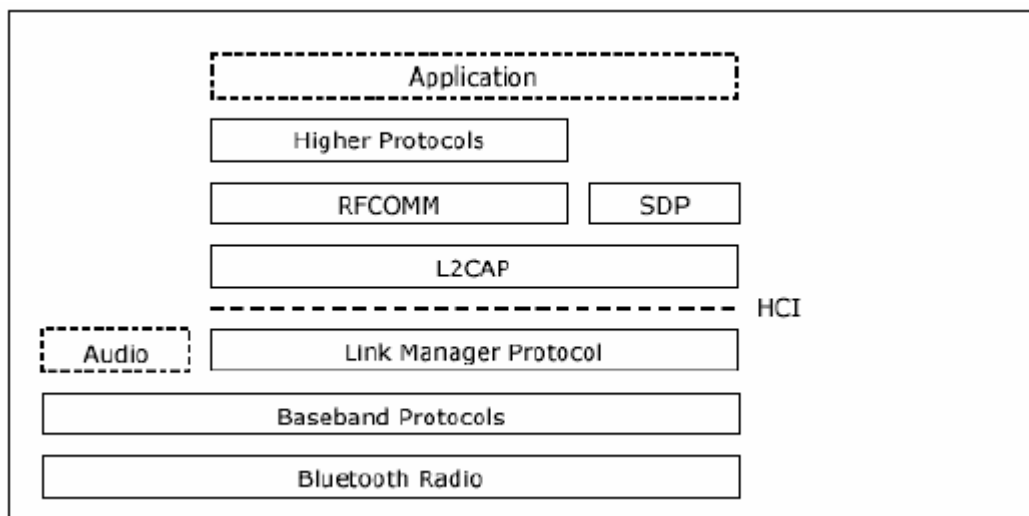
Το εύρος ζώνης (bandwidth) του Bluetooth είναι μόνο 1 MHz ανά κανάλι. Αυτό δεν είναι αρκετό για video εφαρμογές και μερικές άλλες εφαρμογές πολυμέσων. Στο μέλλον το εύρος ζώνης Bluetooth αναμένεται να αυξηθεί.

Ένα άλλο πρόβλημα σχετικά με το Bluetooth σχετίζεται με το μέλλον, όταν η τεχνολογία αυτή θα έχει εξαπλωθεί ευρέως. Τότε πιθανώς να υπάρχουν προβλήματα που σχετίζονται με την περιορισμένη ζώνη συχνοτήτων, ειδικά σε περιοχές που παρατηρείται συνωστισμός, όπως σε έναν αερολιμένα. Το μέγιστο πλήθος από piconets μέσα σε ένα δωμάτιο με μια αποδεκτή υποβάθμιση στην απόδοση είναι σήμερα 10 piconets. Η IEEE εργάζεται στην παραγωγή του προτύπου Bluetooth ως επίσημο IEEE πρότυπο που θα καλείται IEEE 802.15, αλλά έχουν διαπιστώσει ασυνέπειες στην προδιαγραφή, γεγονός που καθιστά δύσκολη τη συγγραφή κάποιου σταθερού standard που θα ικανοποιεί κάποια συγκεκριμένα tests.

2 ΑΡΧΙΤΕΚΤΟΝΙΚΗ BLUETOOTH

2.1 Εισαγωγή

Το Bluetooth είναι μια προδιαγραφή lower-layer κατα την άποψη του OSI. Το ακόλουθο σχήμα παρουσιάζει τα κύρια πρωτόκολλα Bluetooth. Τα μέρη-κλειδιά του είναι το επίπεδο Bluetooth radio (RF), το baseband και το link layer (συμπεριλαμβανομένου του link manager, του logical link control και του πρωτόκολλου προσαρμογής “L2CAP”). Προκειμένου να χρησιμοποιηθούν άλλες εφαρμογές, όπως η μεταφορά αρχείων, η φωνή και το WAP, πάνω από το Bluetooth, περιλαμβάνονται και ανώτερα πρωτόκολλα. Η στοίβα πρωτοκόλλων του Bluetooth παρατίθεται παρακάτω:



ΕΙΚΟΝΑ 2.1

2.2 Bluetooth radio

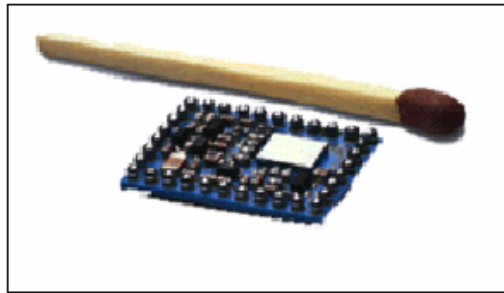
Το επίπεδο αυτό δημιουργεί τις φυσικές συνδέσεις μεταξύ των συσκευών. Για να λειτουργήσει το Bluetooth radio όπως αναφέραμε χρησιμοποιείται παγκοσμίως μια ζώνη συχνοτήτων που καλείται βιομηχανική, επιστημονική και ιατρική ζώνη (ISM band). Η ISM ζώνη εκτείνεται από τα 2.4465 έως τα 2.4835 GHz και χωρίζεται σε 23 κανάλια του 1MHz με το καθένα να υποστηρίζει

μέγιστο ρυθμό ανταλλαγής δεδομένων 1 Mega-symbol ανά δευτερόλεπτο. Το Bluetooth στέλνει κάθε φορά ένα πακέτο σε μια τέτοια σχισμή εύρους 1MHz και έπειτα ο πομπός και ο δέκτης συντονίζονται και οι δύο σε διαφορετικό κανάλι. Αυτό έχει σαν αποτέλεσμα μια περιοδική διαδικασία που εξασφαλίζει ότι χρησιμοποιούνται συνέχεια όλα τα διαθέσιμα κανάλια. Η συμπεριφορά αυτή ονομάζεται **FHSS – Frequency Hopping Spread Spectrum**. Στη μετάδοση με Bluetooth γίνονται 1600 εναλλαγές το δευτερόλεπτο μεταξύ των διαθέσιμων καναλιών (hop rate), που σημαίνει ότι κάθε χρονική στιγμή διαρκεί 625 microseconds , χρόνος που επαρκεί για την αποστολή ενός πακέτου. Βέβαια υπάρχει και η δυνατότητα να στείλουμε μεγαλύτερα πακέτα που ο χρόνος αποστολής τους διαρκεί πάνω από 3 ή και 5 χρονικές σχισμές με αποτέλεσμα να επιτυγχάνουμε μεγαλύτερες ταχύτητες μεταφοράς δεδομένων.

Ο λόγος για την υλοποίηση αυτής της γρήγορης εναλλαγής καναλιών και του μικρού μεγέθους των πακέτων είναι η δυνατότητα λειτουργίας στην ίδια συχνότητα με όλες τις άλλες τεχνολογίες που χρησιμοποιούν αυτήν την ζώνη καθώς και η αύξηση της αξιοπιστίας στη μεταφορά των δεδομένων. Για παράδειγμα, αν ένα τμήμα της ISM μάλιστα δέχεται ισχυρές παρεμβολές, με αποτέλεσμα να δυσχεραίνει ή και να κάνει αδύνατη τη μετάδοση στη συγκεκριμένη συχνότητα, άλλες περιοχές σίγουρα θα είναι καθαρές και η επαναμετάδοση μετά από κάποιες εναλλαγές συχνότητας θα καταστεί δυνατή με αποτέλεσμα να συνεχιστεί η επικοινωνία.

Όπως αναφέραμε η ISM ζώνη είναι ελεύθερη, παρόλα αυτά στην Γαλλία και σε ορισμένες λίγες ακόμα χώρες, δεν επιτρέπεται η χρήση των ιδίων συχνοτήτων, και έτσι έχουν μια στενότερη ζώνη συχνότητας. Παράλληλα ένας ειδικός αλγόριθμος frequency hopping έχει δημιουργηθεί για να συμμορφωθεί με αυτούς τους περιορισμούς. Εντούτοις, τα προϊόντα που χρησιμοποιούν αυτόν τον αλγόριθμο δεν μπορούν να αλληλεπιδράσουν με τα προϊόντα που χρησιμοποιούν τον τυποποιημένο αλγόριθμο frequency hopping. Προβλέπεται, ότι η Γαλλία και οι υπόλοιπες χώρες σύντομα θα αλλάξουν τη ζώνη συχνότητάς, ώστε να συμμορφώνεται με τα παγκόσμια πρότυπα.

Παρακάτω δίνεται ένα σχήμα που δείχνει χαρακτηριστικά το μέγεθος ενός Bluetooth Radio και το πόσο μικρό είναι αυτό σε σχέση με ένα σπύρτο.



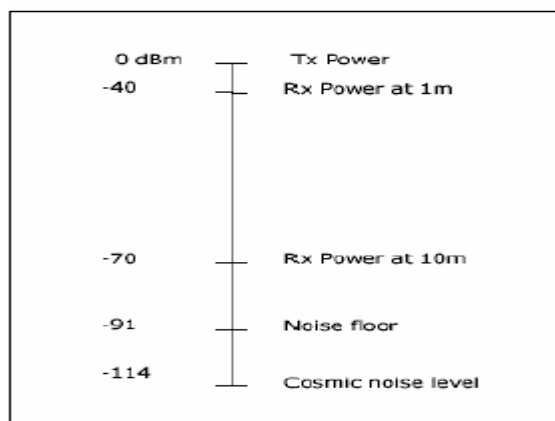
To Bluetooth Radio

ΕΙΚΟΝΑ 2.2

2.3 Χαρακτηριστικά πομπού

Το εύρος των καναλιών είναι 1 MHz ενώ για τα 79 κανάλια (23 για τους ειδικούς αλγορίθμους) χρησιμοποιείται μια εμβέλεια guard στην κορυφή και στο κατώτατο σημείο της ζώνης συχνότητας (frequency band). Η εμβέλεια είναι περίπου 10 μέτρα για χρησιμοποιούμενη ονομαστική ισχύ ίση με 0 dBm, ενώ μπορεί να επεκταθεί στα 250 μέτρα με αύξηση της ισχύος στα 20 dBm. Ο εξοπλισμός των συσκευών - πομπών μπορεί να ανήκει σε μια από τρεις κατηγορίες ισχύος:

- ✦ Class 1 - Η κατηγορία 1 έχει μια μέγιστη ισχύ παραγωγής 100 mW (20 dBm), και μια ελάχιστη ίση με 1 mW (0 dBm).
- ✦ Class 2 - Η κατηγορία 2 έχει μια μέγιστη ισχύ παραγωγής 2.5 mW (4 dBm), και μια ελάχιστη 0.25 mW (-6 dBm), ενώ η ονομαστική ισχύς παραγωγής της είναι 1 mW (0 dBm).



Εικόνα 2.3

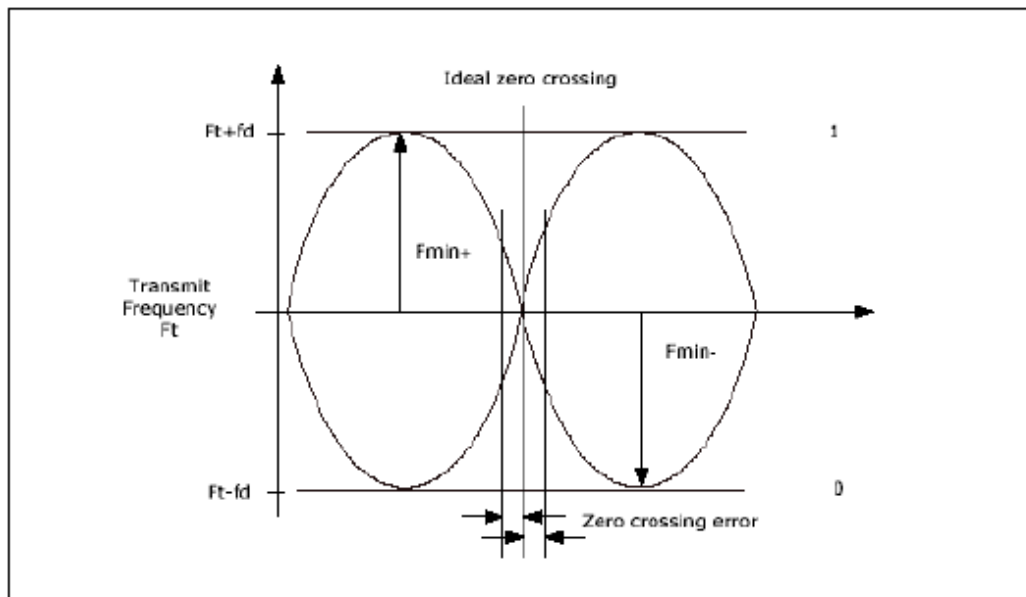
- ✦ Class 3 – Η κατηγορία 3 έχει μια μέγιστη ισχύ παραγωγής 1 mW (0 dBm).

Η κατηγορία 1 απαιτεί ως εξοπλισμό ένα control ισχύος, για να περιοριστεί η ισχύς που παράγεται σε επίπεδα ψηλότερα των 0 dBm. Ένα προϊόν κατηγορίας 2 πρέπει να είναι σε θέση να περιορίσει την ισχύ μετάδοσης το πολύ στα 4 dBm. Η ύπαρξη του control ισχύος δεν είναι υποχρεωτική για τα προϊόντα της κατηγορίας 2 και 3.

2.3.1 Χαρακτηριστικά διαμόρφωσης

Η τεχνική της διαμόρφωσης που χρησιμοποιείται είναι η δυαδική διαμόρφωση μετατόπισης συχνότητας – Binary Frequency Shift Keying (BFSK). Ένα δυαδικό 1 αντιπροσωπεύεται από μια θετική απόκλιση της συχνότητας, ενώ ένα δυαδικό 0 αντιπροσωπεύεται από μια αρνητική απόκλιση της συχνότητας.

Η ελάχιστη απόκλιση της συχνότητας (FD) για μια γρήγορα μεταβαλλόμενη ακολουθία (π.χ. 1010) δεν πρέπει να είναι μικρότερη από το 80% της απόκλισης της συχνότητας για μια αργά μεταβαλλόμενη ακολουθία (π.χ. 00001111). Επίσης η ελάχιστη απόκλιση της συχνότητας δεν πρέπει να είναι μικρότερη από 115 KHz. Το λάθος zero crossing πρέπει να είναι λιγότερο από το $\pm 1/8$ ενός συμβόλου. Ο δέκτης έχει ένα ποσοστό λάθους bit λιγότερο ή ίσο με το 0.1% της ισχύς εισόδου 20dBm.



Πραγματική διαμόρφωση μετάδοσης

EIKONA 2.3.1

2.3.2 Baseband

Το επίπεδο αυτό ουσιαστικά ελέγχει το επίπεδο ραδιοζεύξεων. Συγκεκριμένα, καθορίζει την ακολουθία των αλμάτων της συχνότητας (FH), πραγματοποιεί κρυπτογράφηση κατώτερου επιπέδου για ασφάλεια των ζεύξεων και διαχειρίζεται τα πακέτα δεδομένων όσον αφορά τον έλεγχο σφαλμάτων και την επαναμετάδοση. Επίσης, στο επίπεδο αυτό πραγματοποιούνται οι συνδέσεις και γίνεται ο έλεγχος των διευθύνσεων των γειτονικών συσκευών, όπως θα δούμε σε επόμενη παράγραφο, που αναφέρεται στην εγκατάσταση συνδέσεων. Επιπρόσθετα, υποστηρίζονται δύο ειδών υπηρεσίες:

- Υπηρεσίες με Σύνδεση (SCO): Χρησιμοποιούνται για σύγχρονα δεδομένα όπως φωνή.
- Υπηρεσίες χωρίς Σύνδεση (ACL): Χρησιμοποιούνται για εφαρμογές μεταφοράς δεδομένων τα οποία δεν απαιτούν σύγχρονες ζεύξεις.

2.3.3 Piconets και Scatternets

Δύο ή περισσότερες μονάδες Bluetooth που μοιράζονται το ίδιο κανάλι διαμορφώνουν ένα piconet. Η μια, και μόνο η μία μονάδα, πρέπει να είναι master του piconet ενώ οι άλλες slaves. Τα slaves μπορούν να συμμετέχουν σε διάφορα piconets, ως slaves ή master. Εντούτοις, μόνο μια μονάδα Bluetooth μπορεί να είναι master ενός piconet. Δύο ή περισσότερα piconets που έχουν καλύψει τις γεωγραφικές περιοχές κάλυψης διαμορφώνουν ένα scatternet. Τα μεμονωμένα piconets παραμένουν μη συγχρονισμένα και διατηρούν το hopping συχνότητάς τους.

2.3.4 Φυσικά κανάλια και συνδέσεις

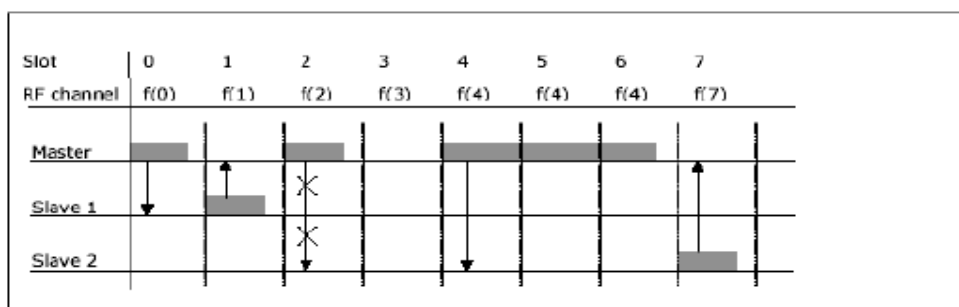
Ένα φυσικό κανάλι piconet καθορίζεται από τον master που αρχικοποιεί μια hopping ακολουθία μέσω των διαθέσιμων καναλιών RF. Γενικά 79 κανάλια RF είναι διαθέσιμα, αλλά σε μερικές χώρες η ζώνη συχνότητας είναι στενότερη και μόνο 23 κανάλια RF μπορούν να χρησιμοποιηθούν. Το κανάλι διαιρείται σε 625 χρονικές σχισμές όπου κάθε σχισμή αντιπροσωπεύει μια συχνότητα RF hop. Σε ειδικές περιπτώσεις, η συχνότητα hop μπορεί να είναι η ίδια για τρεις ή πέντε διαδοχικές χρονικές σχισμές. Οι χρονικές σχισμές διαιρούνται σε δύο κατηγορίες: Σε σχισμές master-slaves, όπου ο master έχει την άδεια για να στείλει στα slaves και στις σχισμές slaves-master όπου ο slave έχει την άδεια για να στείλει στον master. Στις σχισμές s-M, μόνο ο slave που εξετάζεται στην προηγούμενη σχισμή m-S έχει την άδεια να στείλει. Αυτή η μορφή αμφίδρομης μετάδοσης καλείται TDD (Time division Duplex).

Το πρότυπο Bluetooth καθορίζει στις αναφέραμε δύο τύπους φυσικών συνδέσεων: Synchronous Connection Oriented (SCO) και Asynchronous Connection Less (ACL). Γενικά, οι συνδέσεις SCO χρησιμοποιούνται για τις συνδέσεις μετάδοσης φωνής και οι ACL για τη μετάδοση δεδομένων και τη σηματοδότηση.

Κατά την εγκατάσταση μιας SCO σύνδεσης, ο master διατηρεί χρονικές σχισμές (time slots) σε κανονικά χρονικά διαστήματα για την κυκλοφορία SCO δεδομένων σε έναν συγκεκριμένο slave. Παραδείγματος χάριν, μπορεί να

διατηρήσει κάθε $4^{\text{η}}$ m-s χρονική σχισμή (time slot) για τη σύνδεση SCO με αριθμό slave ίσο με τρία. Ο slave έχει την άδεια για να στείλει στον master στις ακόλουθες slave-master στιγμές εκτός αν στις slave διευθυνσιοδοτήθηκε στην προηγούμενη στιγμή master-slave. Η σύνδεση SCO είναι συμμετρική, δηλαδή το bit rate είναι το ίδιο και στις δύο κατευθύνσεις. Προφανώς, η σύνδεση SCO είναι τύπου από σημείο σε σημείο και μπορεί να θεωρηθεί ως circuit-switched σύνδεση. Καμία ικανότητα διόρθωσης λάθους δεν είναι διαθέσιμη.

Το ACL είναι, σε αντίθεση με τη σύνδεση SCO, packet-switched και υποστηρίζει ασύγχρονες και ισόχρονες υπηρεσίες. Οι συνδέσεις ACL μπορούν να είναι συμμετρικές ή αντισυμμετρικές. Οι πρώτες τέσσερις χρονικές σχισμές στο παρακάτω σχήμα παρουσιάζουν μια επιτυχή μετάδοση που ακολουθείται από μια ανεπιτυχή (για την διόρθωση λαθών χρησιμοποιείται η επαναμετάδοση πακέτων). Τα πακέτα ACL μπορούν να είναι broadcast ή διευθυνσιοδοτημένα για έναν συγκεκριμένο slave.



Στο σχήμα αυτό δείχνεται πώς μεταδίδονται τα πακέτα σε ένα Bluetooth piconet.

Τα 4 πρώτα slots (χρονικές στιγμές) δείχνουν μια επιτυχημένη μετάδοση ακολουθούμενη από μια ανεπιτυχή. Ο Slave 2 δεν επιτρέπεται να στείλει αν δεν διαβάσει την διεύθυνσή του στην επικεφαλίδα του πακέτου. Στα slots 4 και 7, δείτε πως το ίδιο RF κανάλι χρησιμοποιείται κατά τη διάρκεια της μετάδοσης ολόκληρου του πακέτου

ΕΙΚΟΝΑ 2.3.4

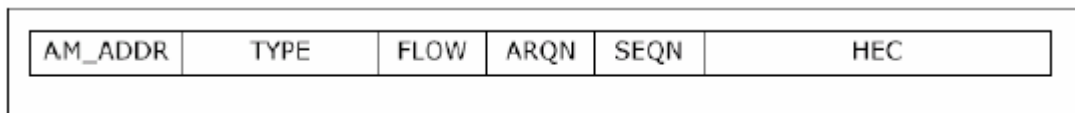
2.3.5 Κωδικός πρόσβασης πακέτων

Για την αναγνώριση DC offset και συγχρονισμού, όλα τα πακέτα έχουν έναν κωδικό πρόσβασης. Ο κωδικός πρόσβασης είναι τα πρώτα 72 ή 68 bits ενός πακέτου. Η περίπτωση των 68 bits ισχύει μόνο εάν δεν υπάρχει καμία επικεφαλίδα πακέτου. Τρεις τύποι κωδικών πρόσβασης καθορίζονται στην προδιαγραφή:

- ✦ Κωδικός πρόσβασης καναλιών – Channel Access Code: Αυτός ο κωδικός συμπεριλαμβάνεται σε όλα τα πακέτα που διαβιβάζονται σε ένα piconet ενώ προσδιορίζει σε πιο piconet ανήκουν τα πακέτα.
- ✦ Κωδικός πρόσβασης συσκευών – Device Access Code: Όταν ο master επιθυμεί να σελιδοποιήσει (page) έναν slave, χρησιμοποιεί τον κωδικό πρόσβασης συσκευών. Αυτός ο κωδικός χρησιμοποιείται επίσης και για άλλους σκοπούς σηματοδότησης.
- ✦ Κωδικός πρόσβασης έρευνας – Inquiry Access Code: Αυτός ο κωδικός χρησιμοποιείται όταν μια συσκευή επιθυμεί να ανακαλύψει άλλες συσκευές. Υπάρχουν δύο τύποι των IACs. Γενικά μπορεί να χρησιμοποιηθεί εάν μια συσκευή επιθυμεί να ανακαλύψει όλες τις συσκευές Bluetooth στην εμβέλεια της ή αν επιθυμεί να βρει συσκευές με συγκεκριμένα χαρακτηριστικά.

2.3.6 Επιγραφή πακέτων

Η επιγραφή πακέτων περιλαμβάνει έξι πεδία, όπως φαίνεται και στο σχήμα που ακολουθεί:



Το format της επικεφαλίδας του πακέτου

ΕΙΚΟΝΑ 2.3.6

Τα πεδία αυτά είναι:

- ✦ Η ενεργός διεύθυνση των μελών – Active Member address (3 bits): Το AM_ADDR διευκρινίζει σε ποιο ενεργό μέλος ανήκει το πακέτο. Τα τρία bit δημιουργούν οκτώ συνδυασμούς, έναν για κάθε ενεργό slave (μόνο επτά slaves μπορούν να είναι ενεργοί συγχρόνως) και έναν για το broadcast. Ο master δεν χρειάζεται κανένα AM_ADDR δεδομένου ότι όλα τα πακέτα στις χρονικές στιγμές slave-to-master προορίζονται για αυτόν. Έτσι χρησιμοποιεί το AM_ADDR για να διακρίνει ποιος slave έστειλε το πακέτο. Η ενεργός διεύθυνση των μελών είναι προσωρινή, π.χ. ένας slave που είναι σε κατάσταση park ορίζει ένα νέο AM_ADDR όταν γίνεται πάλι ενεργός.
- ✦ Κώδικας τύπων – Type Code (4 bits): Το πεδίο TYPE διευκρινίζει, μεταξύ άλλων, με ποιο είδος σύνδεσης διαβιβάζεται το πακέτο (SCO ή ACL) και για πόσες χρονικές σχισμές θα διαρκέσει.
- ✦ Έλεγχος ροής (1 bit) – Flow Control: Το bit FLOW γίνεται “1” όταν μια συσκευή έχει γεμίσει τον buffer λήψης. Όταν ο buffer αδειάζει, το bit FLOW τίθεται πάλι στο “0” και τα πακέτα ACL μπορούν να παραληφθούν πάλι. Τα πακέτα SCO διαβιβάζονται ανεξάρτητα από το bit FLOW.
- ✦ Αναγνώριση της ένδειξης – Acknowledge Indication (1 bit): Αυτό το bit τίθεται εάν η υποδοχή του προηγούμενου πακέτου ήταν επιτυχής. Το CRC χρησιμοποιείται για τον έλεγχο λάθους.
- ✦ Αριθμός ακολουθίας – Sequence Number (1 bit): Το bit SEQN είναι ανιχνεύσιμο για κάθε πακέτο που στέλνεται με το CRC. Αυτό καθιστά εύκολο για τον μεταδότη να διακρίνει ποιο είναι το πακέτο που διαβιβάστηκε/παραλήφθηκε ανεπιτυχώς όταν έλαβε ένα προδιορισμένο κομμάτι ARQN.
- ✦ Έλεγχος λάθους επιγραφών – Header Error Check (8 bits): Ο HEC συμπεριλαμβάνεται για να εξασφαλίσει την ακεραιότητα των επικεφαλίδων.

2.3.7 Packet payload format

Το ωφέλιμο φορτίο (payload) ενός πακέτου διαιρείται σε δύο πεδία: data και voice. Τα πακέτα ACL έχουν μόνο ένα πεδίο δεδομένων και τα πακέτα SCO μόνο ένα πεδίο voice. Υπάρχει επίσης ένα ειδικό πακέτο που περιέχει data και voice παράλληλα.

Το πεδίο δεδομένων έχει μια επικεφαλίδα ωφέλιμων φορτίων. Αυτή η επικεφαλίδα προσδιορίζει το λογικό κανάλι, ελέγχει τη ροή στα λογικά κανάλια και έχει έναν δείκτη μήκους ωφέλιμων φορτίων. Το σώμα ωφέλιμων φορτίων περιέχει τα δεδομένα χρηστών και το μήκος του είναι αποθηκευμένο στο δείκτη μήκους ωφέλιμων φορτίων. Το ωφέλιμο φορτίο των πακέτων ACL προστατεύεται από τα λάθη μετάδοσης από έναν δεκαεξάμπιτο κώδικα CRC.

2.4 Πρωτόκολλο-link manager (LMP)

Το πρωτόκολλο link manager (LMP) παρέχει τα μέσα για την ασφαλή εγκατάσταση συνδέσεων για voice και data. Έχει τη δυνατότητα, όποτε κρίνεται απαραίτητο, να ενημερώνει τις ιδιότητες της σύνδεσης, ώστε να εξασφαλίζει τη βέλτιστη απόδοση. Ο link manager ολοκληρώνει επίσης τις συνδέσεις, είτε έπειτα από αίτημα των υψηλότερων στρωμάτων, είτε λόγω των διαφόρων αποτυχιών.

Το LMP είναι ένα από τα δύο πρωτόκολλα σε επίπεδο συνδέσεων. Χειρίζεται την **ασφάλεια συνδέσεων**, π.χ. κλειδιά, ένωση, επικύρωση και κρυπτογράφηση, **τις πληροφορίες συγχρονισμού**, π.χ. ρολόι, offset και συγχρονισμό των σχισμών ακρίβειας και **τις καταστάσεις (modes)**, π.χ. ενεργό, hold, sniff και park. Το πρωτόκολλο link manager παρέχει επίσης έναν μηχανισμό για το QoS (ποιότητα της υπηρεσίας) και το RSSI (λαμβανόμενη ένδειξη ισχύος σημάτων).

Οι εντολές των link manager στέλνονται ως ωφέλιμο φορτίο των μηνυμάτων υψηλής προτεραιότητας. Οι εντολές αυτές δεν χρειάζονται καμία επιβεβαίωση δεδομένου ότι τα χαμηλότερα στρώματα παρέχουν μια αξιόπιστη σύνδεση. Δύο μηνύματα απάντησης καθορίζονται. Και τα δύο περιέχουν πληροφορίες σχετικά με τον αν και ποιο μήνυμα έγινε αποδεκτό ή μη αποδεκτό.

Στην τελευταία περίπτωση μάλιστα διευκρινίζεται γιατί το μήνυμα δεν έγινε αποδεκτό.

2.4.1 Ασφάλεια

Με τη χρησιμοποίηση των τεχνικών επικύρωσης και κρυπτογράφησης, το Bluetooth παρέχει στο χρήστη ασφαλείς συνδέσεις. Δύο συσκευές λαμβάνουν μέρος στην διαδικασία: Ο ελεγκτής, ο οποίος είναι αυτός που αρχικοποιεί τη διαδικασία επικύρωσης και ο ενάγοντας. Η διαδικασία επικύρωσης μεταξύ δύο συσκευών Bluetooth μπορεί να γίνει με δύο διαφορετικούς τρόπους:

1. Ο ενάγων συνδέει ένα κλειδί συνδέσεων με τον ελεγκτή: Μια απλή επικύρωση γίνεται σε δύο στάδια. Ο ελεγκτής στέλνει έναν τυχαίο αριθμό στον ενάγοντα που υπολογίζει μια απάντηση και την στέλνει πίσω. Εάν η απάντηση είναι σωστή η επικύρωση είναι επιτυχής.
2. Ο ενάγων στον δεύτερο τρόπο δεν συνδέει ένα κλειδί συνδέσεων με τον ελεγκτή: Αντίθετα οι δύο συσκευές πρέπει να περάσουν από μια διαδικασία “ζεύγους”. Ένα κλειδί έναρξης δημιουργείται από ένα pin ή έναν τυχαίο αριθμό. Η απάντηση του ελεγκτή υπολογίζεται χρησιμοποιώντας το κλειδί έναρξης αντί ενός κλειδιού συνδέσεων. Εάν το αποτέλεσμα είναι σωστό η επικύρωση είναι επιτυχής. Όταν η επικύρωση ολοκληρωθεί, μπορεί να χρησιμοποιηθεί κρυπτογράφηση. Μια συσκευή έχει διαφορετικές παραμέτρους κρυπτογράφησης για διαφορετικές συσκευές. Εντούτοις, ο master μπορεί να δημιουργήσει ένα προσωρινό κοινό κλειδί συνδέσεων για ένα ολόκληρο piconet εάν θέλει να μεταδώσει κρυπτογραφημένες πληροφορίες στην radio zone.

2.4.2 Τρόποι – Modes

Μια συσκευή Bluetooth (ή μια σύνδεση μεταξύ δύο συσκευών) μπορεί να βρίσκεται σε τέσσερα διαφορετικά modes. Κάθε συσκευή μπορεί να χρησιμοποιήσει διαφορετικά modes για διαφορετικά piconets, αλλά μπορεί να είναι σε ενεργή κατάσταση μόνο σε ένα piconet κάθε χρονική στιγμή. Οι καταστάσεις (modes) έχουν διαφορετική κατανάλωση ισχύος ενώ παρακάτω παρατίθενται κατά φθίνουσα κατανάλωση. Σε μια τυχαία στιγμή, η συσκευή

πρέπει να βρίσκεται σε μια και μόνο μια από αυτές τις καταστάσεις ανά ένα piconet:

1. Ενεργό: Η συσκευή συμμετέχει στο piconet ακούγοντας (timeslots master-to-slaves) τα πακέτα που περιέχουν τα AM_ADDR της (ενεργός διεύθυνση μελών).
2. Sniff: Όταν η συσκευή βρίσκεται σε αυτήν την κατάσταση ενεργεί περίπου όπως και στην ενεργή κατάσταση. Η διαφορά είναι ότι κατά την είσοδο στην κατάσταση sniff, οι master και slave αποφασίζουν για ένα sniff διάστημα, το οποίο είναι ο χρόνος μεταξύ δύο timeslots όπου ο slave παρακολουθεί τα πακέτα.
3. Hold: Όταν μια συσκευή βρίσκεται σε κατάσταση hold τα πακέτα ACL δεν υποστηρίζονται αλλά τα πακέτα SCO μπορούν ακόμα να διαβιβαστούν. Δεν καθορίζεται τι κάνει ο slave κατά τη διάρκεια του χρόνου hold π.χ. μπορεί να ενώνει άλλα piconets ή να κλείνει τον πομποδέκτη του για να εξοικονομήσει ενέργεια. Εντούτοις, εάν μια συσκευή καθιερώσει μια σύνδεση SCO, επιτρέπεται να συμπεριφέρεται έτσι μόνο για μη reserved timeslots.
4. Park: Όταν ο slave βρίσκεται σε κατάσταση Park, σταματά το AM_ADDR του αλλά παραμένει συγχρονισμένος. Του δίνεται μια PM_ADDR (σταθμευμένη διεύθυνση μελών) την οποία ο master χρησιμοποιεί για unparking του slave και μια AR_ADDR (διεύθυνση αιτήματος πρόσβασης) την οποία ο slave χρησιμοποιεί για αίτηση unparked. Ο master μπορεί επίσης να χρησιμοποιήσει το BD_ADDR (διεύθυνση συσκευών Bluetooth) για αυτό. Εάν ο slave θέλει να αφήσει την κατάσταση park, μπορεί να στείλει ένα αίτημα στον master. Το Unparking μπορεί να γίνει σε ορισμένες χρονικές σχισμές, οι οποίες εμφανίζονται σε ένα σταθερό διάστημα (διάστημα αναγνωριστικών σημάτων).

Στον παρακάτω πίνακα ακολουθεί μια επίδειξη του τι επιτρέπεται να κάνουν οι slaves στα διάφορα modes. Στο συγκεκριμένο παράδειγμα, ο slave που τοποθετείται σε Hold Mode έχει ένα SCO link.

| Slot | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---------------|-----------------|------------|------------|------------|------------|------------|-----------------|------------|------------|
| Master | SCO TX | SCO RX | TX | RX | TX | RX | SCO TX | SCO RX | TX |
| Slave, Active | (RX) | (TX) | RX | TX | RX | TX | (RX) | (TX) | RX |
| Slave, Sniff | RX | TX | Un-defined | Un-defined | RX | TX | Un-defined | Un-defined | RX |
| Slave, Hold | SCO RX | SCO TX | Un-defined | Un-defined | Un-defined | Un-defined | SCO RX | SCO TX | Un-defined |
| Slave, Park | Unpark possible | Un-defined | Un-defined | Un-defined | Un-defined | Un-defined | Unpark possible | Un-defined | Un-defined |

Sniff interval: 4 slots
Beacon interval: 6 slots

EIKONA 2.3.10

2.4.3 Σύγχρονες συνδέσεις Connection - Oriented

Όταν μια SCO σύνδεση καθιερώνεται μεταξύ δύο συσκευών, γίνεται κράτηση timeslots. Ένα SCO link δεσμεύει slots περιοδικά, με ένα διάστημα αποκαλούμενο διάστημα SCO. Σε όλες τις συνδέσεις αυτού του είδους ορίζεται μια hold SCO που την προσδιορίζει.

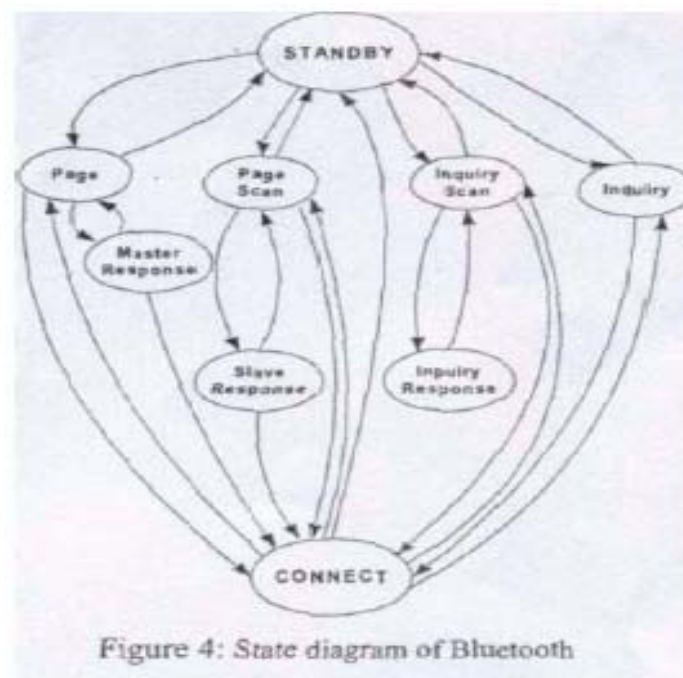
Είτε ο master, είτε ο slave, μπορεί να ζητήσει την εγκατάσταση μιας σύνδεσης SCO. Εάν το ζητήσει ο master, στέλνει ένα σήμα που περιέχει τις προτιμώμενες από αυτόν παραμέτρους της SCO σύνδεσης. Εάν ο slave δεχτεί τις παραμέτρους, θα δημιουργηθεί η σύνδεση SCO, διαφορετικά μπορεί να αποκριθεί διευκρινίζοντας ποια παράμετρο/ους δεν μπορεί να δεχτεί. Ο master έπειτα μπορεί να στείλει ένα νέο αίτημα με άλλες παραμέτρους. Όταν ένας slave ζητά μια σύνδεση SCO, στέλνει ένα σήμα με τις προτιμώμενες παραμέτρους του στον master. Ο master στέλνει έπειτα ένα αίτημα πίσω στο slave όπως περιγράφεται ανωτέρω. Οι παράμετροι πρέπει να είναι οι ίδιες με τις ζητούμενες από τον slave.

2.4.4 Στάδια του Bluetooth - Καθιέρωση της σύνδεσης

Εάν τα υψηλότερα στρώματα της συσκευής σελιδοποίησης χρειαστούν μια σύνδεση, θα σταλεί ένα αίτημα. Η απομακρυσμένη συσκευή μπορεί έπειτα να το δεχτεί ή να το απορρίψει. Εάν το δεχτεί, και οι δύο συσκευές θα κινήσουν τις διαδικασίες οργάνωσης σύνδεσης. Η συσκευή που έχει τελειώσει πρώτη στέλνει ένα πλήρες set-up σήμα και αναμένει ένα άλλο. Όταν και οι δύο τελειώσουν μαζί και τα waiting party λάβουν το πλήρες set-up σήμα, τα none-LMP-packets μπορούν να διαβιβαστούν πέρα από το λογικό κανάλι.

Αναλυτικά :

Τα διάφορα στάδια λειτουργίας των μονάδων Bluetooth καθορίζονται για να υποστηρίξουν τις λειτουργίες εγκατάστασης του καναλιού σε ένα piconet και της προσθήκης και απελευθέρωσης των μονάδων από το piconet. Το σχήμα 4 παρουσιάζει ένα διάγραμμα σταδίων που επεξηγεί τα διαφορετικά στάδια που χρησιμοποιούνται στο Bluetooth.



ΕΙΚΟΝΑ 2.4.4α

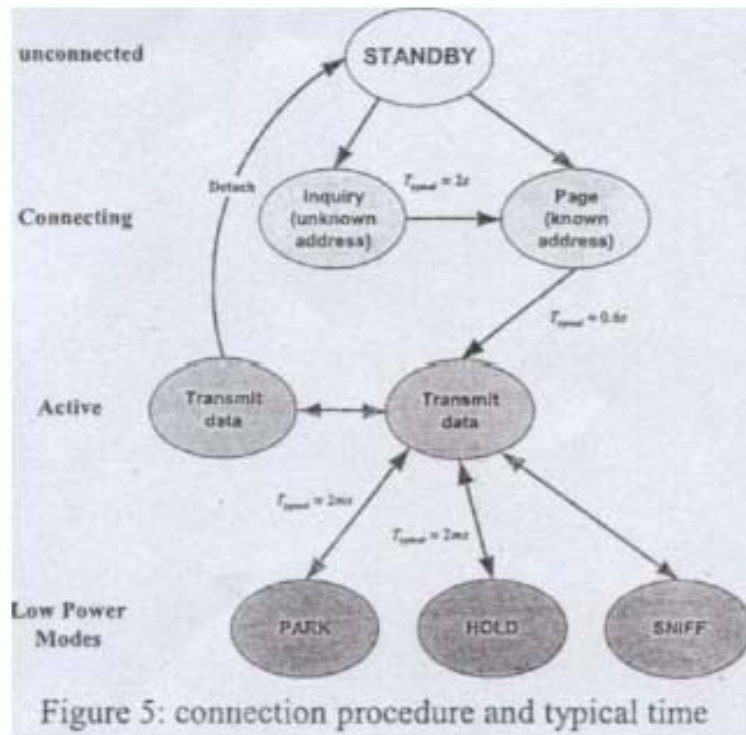
Υπάρχουν δύο σημαντικά στάδια: το STANDBY και το CONNECTION. Επιπρόσθετα υπάρχουν επτά υποστάδια, η σελίδα(page), η ανίχνευση σελίδων (page scan), η έρευνα (inquiry), η ανίχνευση έρευνας (inquiry scan), η master απάντηση (master response), η απάντηση slave (slave response) και απάντηση

έρευνας (inquiry response). Τα υποστάδια είναι προσωρινά στάδια που χρησιμοποιούνται για την προσθήκη νέων slaves σε ένα piconet. Για να κινηθούν οι συσκευές από το ένα στάδιο στο άλλο, χρησιμοποιούνται είτε οι εντολές από το link manager του Bluetooth, είτε εσωτερικά σήματα στον link controller (όπως το σήμα ώθησης (trigger signal) από correlator και τα timeout σήματα (timeout signals)).

Σε περιβάλλον Bluetooth, όταν μια συσκευή slave βρίσκεται σε ένα CONNECT state και κανένα δεδομένο δεν χρειάζεται να διαβιβαστεί, οι μονάδες μπορούν να τεθούν σε κατάσταση hold όπου τρέχει μόνο ένας εσωτερικός timer. Όταν οι μονάδες βγαίνουν από την κατάσταση hold τότε η μεταφορά δεδομένων μπορεί να ξαναξεκινήσει στιγμιαία. Κατά τη διάρκεια της καταστασης hold η μονάδα slave κρατά την ενεργή διεύθυνση μέλους της (AM_ADDR). Οι μονάδες μπορούν έτσι να παραμείνουν συνδεδεμένες, χωρίς μεταφορά δεδομένων, σε κατάσταση low power. Το hold χρησιμοποιείται χαρακτηριστικά κατά τη σύνδεση διάφορων piconets. Θα μπορούσε επίσης να χρησιμοποιηθεί για τις μονάδες όπου τα δεδομένα χρειάζεται να στέλνονται πολύ σπάνια και η μικρή κατανάλωση ισχύος (low-power consumption) είναι σημαντική. Μια παρόμοια κατάσταση είναι η κατάσταση SNIFF που είναι επίσης μια κατάσταση χαμηλής ισχύος (low power). Στην κατάσταση SNIFF, ο master μπορεί να αρχίσει τη μετάδοση μόνο στις διευκρινισμένες χρονικές σχισμές (time slots), οι οποίες αποκαλούνται sniff σχισμές (slots).

Ο αριθμός των συσκευών σε ένα piconet είναι πραγματικά απεριόριστος. Αυτό ισχύει επειδή υπάρχει η κατάσταση park. Όταν ένας slave δεν χρειάζεται να συμμετέχει στο piconet, αλλά θέλει να παραμείνει συγχρονισμένος στο κανάλι, μπορεί να εισαχθεί στην κατάσταση park που είναι κατάσταση χαμηλής ισχύος, με πολύ μικρή δραστηριότητα.

Το σχήμα 5 παρουσιάζει το διάγραμμα ροής των διαδικασιών σύνδεσης και του χαρακτηριστικού χρόνου που απαιτεί η κάθε μετάβαση. Ο χρόνος μετάβασης από την κατάσταση CONNECT στην κατάσταση PARK είναι τόσο σύντομος που μπορεί να επιτρέψει σε έναν master να συνδέσει περισσότερους από 7 slaves σε ένα piconet.



ΕΙΚΟΝΑ 2.4.4β

2.4.5 Άλλα χαρακτηριστικά γνωρίσματα

- ✦ Εναλλαγή του master-slave ρόλου: Το LMP μπορεί να ζητήσει και να επιτρέψει ο master και ο slave να αλλάξουν ρόλους. Αυτό είναι απαραίτητο δεδομένου ότι η συσκευή σελιδοποίησης γίνεται πάντα master του piconet.

Αναλυτικά :

Στο δίκτυο Bluetooth όλες οι μονάδες (units) είναι peer-units με οφθαλμοφανείς διεπαφές hardware και software εκτός από μια μοναδική 48-bit διεύθυνση. Ο διαχωρισμός master και slave είναι προσωρινός. Ισχύει μόνο κατά τη διάρκεια της περιόδου σύνδεσης (connection period).

Master : Στην έναρξη μιας σύνδεσης, η αρχικοποιημένη μονάδα ορίζεται προσωρινά ως master. Είναι ο master που αρχίζει τη σύνδεση και ελέγχει την κυκλοφορία μέχρι τον μέγιστο αριθμό των επτά μονάδων στον ίδιο χρόνο.

Slave: Οι συσκευές που ένωσαν ένα piconet αρχικοποιούνται από έναν master. Θα χρησιμοποιήσουν το ρολόι του master. Δύο slaves σε ένα piconet

δεν έχουν καμία άμεση σύνδεση. Ο master θα καθορίσει τις επικοινωνίες μεταξύ τους.

Πραγματικά, ο αριθμός συσκευών σε ένα piconet είναι πραγματικά απεριόριστος, αν και ένας master μπορεί να έχει "μόνο" επτά ενεργές συσκευές σε οποιαδήποτε δεδομένη χρονική στιγμή. Η επανασύνδεση γίνεται σχεδόν αμέσως και δεν παρουσιάζει κανένα πρακτικό περιορισμό.

- Αίτημα ονόματος: Κάθε συσκευή Bluetooth έχει ένα φιλικό προς το χρήστη όνομα (ανώτατοι 248 χαρακτήρες). Το LMP παρέχει τα μέσα για να προσεγγιστεί αυτό.
- Αποσύνδεση: Μια σύνδεση μεταξύ δύο συσκευών Bluetooth μπορεί να ολοκληρωθεί οποιαδήποτε στιγμή. Είτε ο master είτε ο slave μπορεί να το κάνει. Οποτεδήποτε συμβεί ολοκλήρωση σύνδεσης (αποσύνδεση), στέλνεται ένα σήμα προς την άλλη συσκευή το οποίο εξηγεί τον λόγο ολοκλήρωσης της σύνδεσης.
- Έλεγχος ισχύος: Εάν ένα σήμα με χαμηλή ισχύ (συγκρινόμενη με την προτιμημένη) παραλαμβάνεται από μια συσκευή, η τελευταία μπορεί να ζητήσει να αυξηθεί η παραγόμενη ισχύς του sending party και αντίστροφα εάν η ισχύς των σημάτων είναι πάρα πολύ υψηλή. Ο master έχει διαφορετική και απολύτως ανεξάρτητη διαβιβασθείς ισχύ για διαφορετικούς slaves.
- Επίβλεψη συνδέσεων: Ο master ελέγχει ένα χρονόμετρο (timer) που χρησιμοποιείται για την επίβλεψη των συνδέσεων για κάθε σύνδεση. Το χρονόμετρο ανιχνεύει εάν οι απομακρυσμένες συσκευές κινούνται εκτός εμβέλειας, παρουσιάζουν μια πτώση ισχύος ή οποιαδήποτε άλλη παρόμοια λανθασμένη λειτουργία. Το LMP χρησιμοποιείται για τον καθορισμό της τιμής timeout.

2.4.6 Συμπέρασμα

Οι ευθύνες του πρωτοκόλλου LMP είναι πολλές και ζωτικής σημασίας για τη λειτουργία του συστήματος. Εφόσον το LMP παρέχει όλες τις ευκολίες που αναφέραμε, οι δυνατότητες των υψηλότερων πρωτοκόλλων είναι απεριόριστες: Είναι δυνατό να εγκατασταθούν, να διατηρηθούν και να αποσυνδεθούν οι ασφαλείς συνδέσεις και για την φωνή και για την κυκλοφορία δεδομένων, με τη μέγιστη απόδοση ισχύος. Είναι επίσης δυνατό να τεθούν τα modes των συνδέσεων και να ζητηθεί μια εναλλαγή των master-slave ρόλων. Αυτές οι λειτουργίες δίνουν το συνολικό έλεγχο του piconet. Είναι λοιπόν πολύ σπουδαίο για έναν υπεύθυνο ανάπτυξης Bluetooth να ξέρει και να είσαι σε θέση να χρησιμοποιήσει τις λειτουργίες που προσφέρει το πρωτόκολλο link manager.

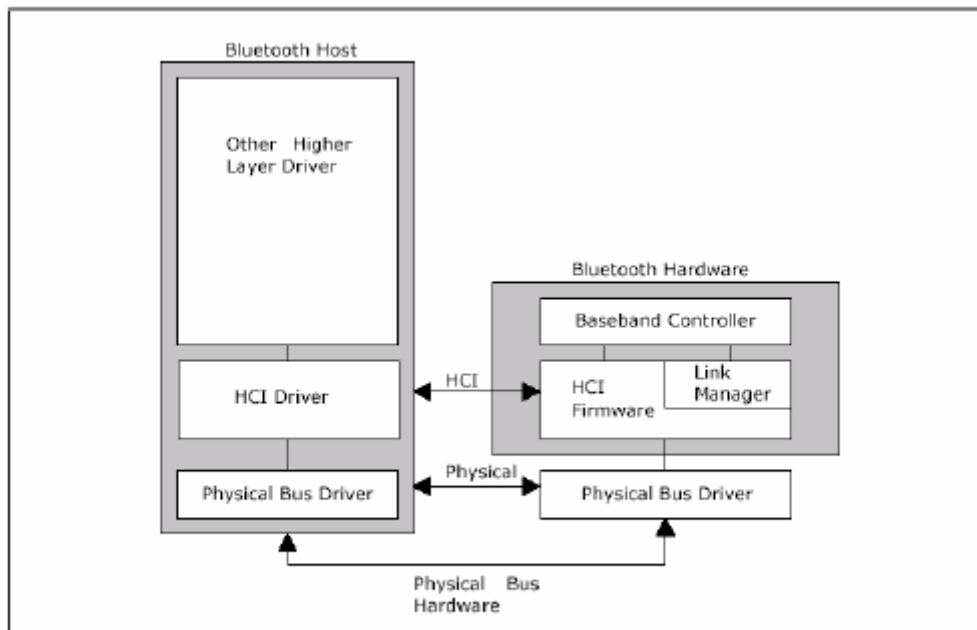
2.5 *HOST CONTROLLER INTERFACE*

Η διεπαφή host controller (HCI) δίνει τη δυνατότητα στα υψηλότερου επιπέδου πρωτόκολλα να έχουν πρόσβαση στα χαμηλότερα στρώματα. Με τη βοήθεια των εντολών που παρέχονται από το HCI, επιτυγχάνεται ο έλεγχος του host baseband και του link manager, καθώς και η πρόσβαση στο hardware status και τους control registers.

2.5.1 Χαμηλότερα επίπεδα και υλικό

Για την πρόσβαση στις εντολές baseband, στις εντολές link manager, στους hardware status registers, στους control registers και στους event registers εφαρμόζονται εντολές HCI. Η ανταλλαγή των εντολών και των δεδομένων γίνεται μεταξύ του οδηγού (driver) HCI στον host και του firmware HCI στο υλικό (hardware) Bluetooth. Το στρώμα μεταφορών host controller παρέχει τη δυνατότητα και στα δύο στρώματα HCI να ανταλλάξουν δεδομένα.

Όταν εμφανίζεται ένα event HCI, ο host προσδιορίζει από πού στρώμα μεταφορών host controller προήλθε. Επίσης αναλύει το πακέτο λήψης για να αποφασίσει ποιο event ήταν.



Περίληψη του Bluetooth Host και του Bluetooth Hardware

EIKONA 2.5.1

Το Bluetooth hardware αποτελείται από ένα αναλογικό μέρος, το radio Bluetooth, και ένα ψηφιακό μέρος, τον host controller (HC). Ο HC έχει ένα ψηφιακό μέρος επεξεργασίας υλικού, τον link controller (LC) και έναν πυρήνα CPU. Χρησιμοποιείται επίσης για την διασύνδεση του περιβάλλοντος host. Ο κύριος ρόλος του link controller είναι να ελέγξει το baseband Bluetooth. Ασχολείται επίσης με την κρυπτογράφηση, την ακουστική κωδικοποίηση κ.λ.π. Αποτελείται από τμήματα υλικού και λογισμικού. Με την βοήθεια του πυρήνα CPU αντιμετωπίζονται οι έρευνες και τα αιτήματα φιλτραρισμού των σελίδων φίλτρων χωρίς την ανάμειξη της συσκευής host. Το λογισμικό LM τρέχει στον πυρήνα του CPU. Το LM ανακαλύπτει άλλα LMs και η επικοινωνία γίνεται μέσω του πρωτοκόλλου link manager (LMP).

Υπάρχει ένα στρώμα μεταφορών μεταξύ του host controller driver και του host controller (π.χ. μια κάρτα PC σε ένα lap-top). Ένα σημαντικό ζήτημα για το στρώμα μεταφορών είναι η διαφάνεια. Παραδείγματος χάριν, ο οδηγός host controller πρέπει να μπορεί να διακρίνει αν χρησιμοποιείται μία κάρτα USB ή μια κάρτα PC.

2.5.2 Εντολές HCI

Οι εντολές αυτές δίνουν στον host τη δυνατότητα να ελέγξει τις συνδέσεις με άλλες συσκευές. Οι link manager ανταλλάσσουν LMP εντολές με τις μακρινές συσκευές Bluetooth. Οτιδήποτε οι HCIs χρησιμοποιούν για να ειδοποιήσουν τον host για την ολοκλήρωση της εντολής, αποτελεί ένα γεγονός (event), όπως π.χ. το link layer status, κ.λ.π. Υπάρχουν επίσης διάφοροι κώδικες λάθους που συνδέονται με τις εντολές HCI. Ακολουθεί μια σύντομη εισαγωγή στις ομάδες εντολών και στις παραμέτρους στο επίπεδο HCI:

- Εντολές control link: Εντολές που δίνουν στον host controller τη δυνατότητα να ελέγξει τις συνδέσεις σε άλλες συσκευές Bluetooth.
- Εντολές συνδέσεων: Εντολές που δίνουν τη δυνατότητα στους host να κατευθύνουν τον τρόπο με τον οποίο ο link manager διαχειρίζεται το piconet. Επίσης, με την χρησιμοποίηση των εντολών αυτών, μπορούν να γίνουν αλλαγές στον τύπο των συνδέσεων (SCO ή ACL).
- Εντολές host controller και Baseband: Αυτές οι εντολές δίνουν πρόσβαση και δυνατότητα ελέγχου του υλικού Bluetooth. Ελέγχουν επίσης τις συσκευές Bluetooth και τις ικανότητες του host controller, του link manager και του Baseband. Ο host μπορεί να χρησιμοποιήσει αυτές τις εντολές για να τροποποιήσει τη συμπεριφορά της τοπικής συσκευής.
- Εντολές testing: Οι εντολές testing χρησιμοποιούνται για τη δοκιμή των διαφορετικών λειτουργιών του υλικού Bluetooth

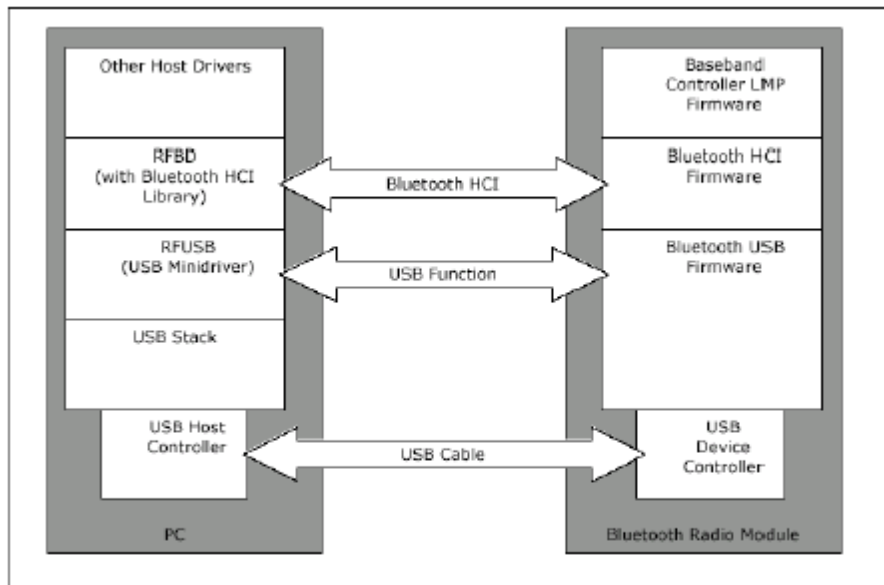
Παράμετροι:

- Ενημερωτικοί παράμετροι: Κάθε bluetooth hardware χαρακτηρίζεται από ορισμένες παραμέτρους που ποικίλουν ανάλογα με τον κατασκευαστή. Οι παράμετροι αυτοί παρέχουν πληροφορίες, τόσο για τη συσκευή Bluetooth, όσο και για τις ικανότητες του host controller, του link manager και της Baseband.
- Παράμετροι θέσης: Οι παράμετροι αυτοί, που τροποποιούνται από τον host controller, παρέχουν πληροφορίες σχετικά με την τρέχουσα κατάσταση του host controller, του link manager και της Baseband.

2.5.3 Επίπεδο μεταφορών (transport layer) HCI USB

Το επίπεδο μεταφορών USB αντιστοιχεί στη διπλής κατεύθυνσης λειτουργία βελών "USB", βλέπε επόμενο σχήμα. Παρέχει τα μέσα να εφαρμοστεί μια συσκευή USB Bluetooth. Το στρώμα μεταφορών HCI USB υποστηρίζει και τα πακέτα ACL και SCO. Επιτρέπει επίσης δυναμική αναγνώριση endpoint (κανάλι) επάνω στην έναρξη οδηγών. Υπάρχουν τέσσερις τύποι endpoint, ελέγχου, διακοπής, συσσώρευσης και ισοχρονισμού. Τα endpoint είναι αντίστοιχα κατάλληλα για να ελέγχουν τις εντολές HCI, να διακόπτουν τα γεγονότα HCI, να συσσωρεύουν τα δεδομένα ACL, να ισοχρονίζουν τα δεδομένα SCO.

- έλεγχος του endpoint, το endpoint 0, χρησιμοποιείται για τη διαμόρφωση και τον έλεγχο της συσκευής USB. Όταν ένα πακέτο παραλαμβάνεται πέρα από αυτό το endpoint, περιέχει τον κώδικα κατηγορίας Bluetooth και αντιμετωπίζεται ως πακέτο εντολής HCI.
- Διακοπή του endpoint εξασφαλίζει ότι τα γεγονότα παραδίδονται σε μια έγκαιρη fashion με την εγγυημένη λανθάνουσα κατάσταση.
- Τα μαζικά endpoint επιτρέπουν στους πολλαπλάσιους buffers των 64-byte να στείλουν πλαίσιο ενός χιλιοστού του δευτερολέπτου, ανάλογα με το διαθέσιμο εύρος ζώνης του bus. Έχει επίσης την ανίχνευση και τη διόρθωση λάθους, η οποία το καθιστά κατάλληλο για τα δεδομένα ACL.
- Ισόχρονα δεδομένα μεταφοράς SCO endpoint από και προς τους host controllers. Το radio είναι ικανό για τρία των 64 kbps καναλιών φωνής που κωδικοποιούνται με τη δεκαεξάμπιτη γραμμική ακουστική κωδικοποίηση.



Σχέση μεταξύ του host και του Bluetooth Radio Module

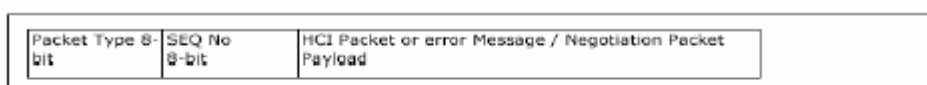
EIKONA 2.5.3

Για να επιτραπεί η φόρτωση του κατάλληλου driver stack, ανεξάρτητα από το ποιος είναι ο κατασκευαστής της συσκευής, θα χρησιμοποιηθεί ένας κώδικας κατηγορίας Bluetooth. Χρησιμοποιείται επίσης για να διαφοροποιήσει τις εντολές USB από τις εντολές HCI.

2.5.4 Στρώμα μεταφορών HCI RS- 232

Ο στόχος του στρώματος μεταφορών RS- 232 είναι να καταστεί δυνατή η χρησιμοποίηση του Bluetooth HCI, μέσω μιας φυσικής διεπαφής RS- 232 μεταξύ του host BT και του host controller BT.

Υπάρχουν τέσσερις τύποι πακέτων HCI που μπορούν να σταλούν μέσω του στρώματος μεταφοράς RS- 232: Πακέτα εντολής HCI, πακέτα γεγονότων HCI, πακέτα ACL HCI και πακέτα HCI SCO. Το HCI δεν διαφοροποιείται ανάλογα με τον τύπο του πακέτου. Επομένως, εάν τα πακέτα HCI στέλνονται μέσω μιας κοινής φυσικής διεπαφής, είναι απαραίτητη η προσθήκη ενός δείκτη πακέτων HCI, ώστε να διαφοροποιούνται οι τέσσερις τύποι πακέτων.



Το βασικό RS232 packet frame

EIKONA 2.5.4

Υπάρχουν δύο πρόσθετοι τύποι πακέτων, που χρησιμοποιούνται για να υποστηρίξουν τη δυναμικές διαπραγματεύσεις και την υποβολή έκθεσης λάθους. Τον δείκτη πακέτων HCI ακολουθεί ένας οκτάμπιτος αριθμός ακολουθίας, ο οποίος αυξάνεται κάθε φορά που στέλνονται οποιαδήποτε από τα παραπάνω πακέτα. Εξαίρεση αποτελεί η αποστολή των πακέτων αναμετάδοσης ως μέρος της αποκατάστασης λάθους. Πριν σταλούν οποιαδήποτε στοιχεία μέσω μιας σύνδεσης RS- 232 το baud rate, ο τύπος ισοτιμίας, ο αριθμός των stop bits και το mode πρωτοκόλλου, αποτελούν αντικείμενα διαπραγμάτευσης ανάμεσα στον host controller και τον host.

2.5.5 Στρώμα μεταφορών HCI UART

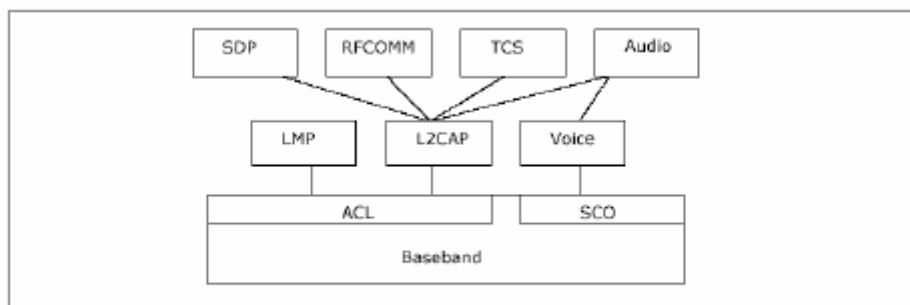
Ο στόχος του στρώματος μεταφορών RS- 232 είναι να καταστεί δυνατή η χρησιμοποίηση του Bluetooth HCI, μέσω μιας σειριακής διεπαφής μεταξύ δύο UARTs, στον ίδιο πίνακα κυκλωμάτων. Υποθέτει ότι δεν υπάρχει κανένα λάθος στην γραμμή και παρουσιάζει πολλές ομοιότητες με το στρώμα μεταφορών HCI RS232. Υπάρχουν τέσσερις τύποι πακέτων HCI που μπορούν να σταλούν μέσω του στρώματος μεταφορών UART: Πακέτα εντολής HCI, πακέτα γεγονότων HCI, πακέτα ACL HCI και πακέτα HCI SCO. Τα πακέτα εντολής HCI μπορούν να σταλούν μόνο στον BT host controller, ενώ τα πακέτα γεγονότων HCI μπορούν να σταλούν μόνο από τον BT host controller. Τα πακέτα HCI στέλνονται μέσω μιας κοινής φυσικής διεπαφής, και ως εκ τούτου πρέπει να προστεθεί ένας δείκτης πακέτων HCI.

2.5.6 Συμπέρασμα

Το HCI είναι η διεπαφή μέσω της οποίας πραγματοποιείται η επικοινωνία με τα χαμηλότερα στρώματα, τα οποία βρίσκονται στο Bluetooth hardware. Μέσω του HCI τα υψηλότερου επιπέδου πρωτόκολλα, που βρίσκονται στον Bluetooth host, μπορούν να έχουν άμεση πρόσβαση στα χαμηλότερα πρωτόκολλα (π.χ. το L2CAP μπορεί να έχει άμεση πρόσβαση στο πρωτόκολλο baseband χωρίς τη συμμετοχή του LMP). Αυτό απλοποιεί τον σχεδιασμό των υψηλότερων επιπέδων και μειώνει την πολυπλοκότητα στα χαμηλότερα επίπεδα.

2.6 Λογικό πρωτόκολλο ελέγχου και προσαρμογής συνδέσεων (L2CAP)

Το λογικό πρωτόκολλο ελέγχου και προσαρμογής συνδέσεων (L2CAP) είναι ένα πρωτόκολλο που βοηθά και υποστηρίζει τα πολύπλοκα πρωτοκόλλα υψηλότερου επιπέδου. Ασχολείται με την κατάτμηση και την επανασυναρμολόγηση πακέτων, τη συντήρηση της ποιότητας υπηρεσίας σε αποδεκτά επίπεδα και τη διαχείριση της ομάδας σε ένα piconet. Το L2CAP βρίσκεται πάνω από το πρωτόκολλο baseband και διασυνδέεται με υψηλότερα πρωτόκολλα όπως SDP, RFCOMM και TCS. Δεν ασχολείται με την προστασία, καθώς όλη η προστασία των μεταφερόμενων πληροφοριών (έλεγχος ακεραιότητας κ.λ.π) γίνεται σε επίπεδο baseband.

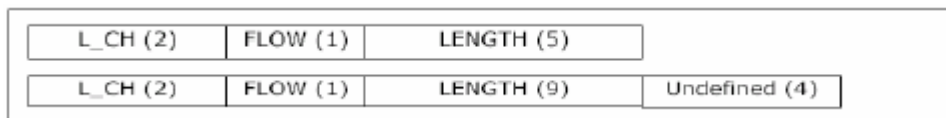


To L2CAP στο protocol stack

ΕΙΚΟΝΑ 2.6

2.6.1 Λειτουργική περιγραφή

Τα πακέτα δεδομένων μέχρι 64 kilobyte μπορούν να διαβιβαστούν ή να παραληφθούν από τα υψηλότερου επιπέδου πρωτόκολλα ή τις εφαρμογές. Υπάρχουν δύο τύποι, single-slot και multi-slot πακέτα L2CAP. Η μόνη διαφορά είναι το μέγεθος στο πεδίο μήκους.



Επικεφαλίδα payload για single-slot πακέτα (πάνω) και multi-slot πακέτα (κάτω)

ΕΙΚΟΝΑ 2.6.1

Ο ελεγκτής (LC) συνδέσεων διαχειρίζεται το FLOW bit. Εάν είναι 1 αντιπροσωπεύει το “flow on”. Όταν καμία κυκλοφορία L2CAP δεν μπορεί να σταλεί, μέσω της σύνδεσης ACL, το bit τίθεται 0. Τα σημαντικά ζητήματα για το L2CAP είναι η απλότητα και τα χαμηλά γενικά έξοδα. Η πολυπλοκότητα του πρωτοκόλλου πρέπει να είναι αποδεκτή σε πολλές διαφορετικές συσκευές όπως τα κυψελοειδή τηλέφωνα, τους προσωπικούς υπολογιστές, τα πληκτρολόγια κ.λ.π.

2.6.2 Κύρια χαρακτηριστικά

Παρακάτω εξετάζονται τα κύρια χαρακτηριστικά του πρωτοκόλλου L2CAP:

- ✦ **Multiplexing πρωτόκολλο:** Το πρωτόκολλο baseband δεν μπορεί να προσδιορίσει τα διάφορα υψηλότερα πρωτόκολλα που είναι πολυπλεγμένα επάνω σε αυτό. Με τη χρησιμοποίηση του L2CAP είναι δυνατό να διακριθούν τα διάφορα ανώτερα πρωτόκολλα στρώματος όπως SDP ή RFCOMM.
- ✦ **Κατάτμηση και επανασυναρμολόγηση (SAR):** Τα πακέτα δεδομένων που καθορίζονται από το πρωτόκολλο baseband είναι περιορισμένα στο μέγεθος. Επομένως τα μεγάλα πακέτα L2CAP πρέπει να περιέχουν τα μικρότερα πακέτα baseband. Η λειτουργία SAR είναι απολύτως απαραίτητη για να υποστηρίξει τα πρωτόκολλα που χρησιμοποιούν

μεγαλύτερα πακέτα από εκείνα που υποστηρίζονται από τη baseband. Το SAR μειώνει το overhead που προκαλείται από την διάδοση στο δικτύο και παρέχει τον τρόπο ώστε να γίνει η μεταφορά των πακέτων που χρησιμοποιούνται από τα πρωτόκολλα πιο υψηλού επιπέδου μέσω των πακέτων baseband.

- ✦ Κατάτμηση: Το L2CAP έχει ένα όριο για το μέγεθος πακέτων του ανώτερου στρώματος, που αποκαλείται μέγιστη μονάδα μετάδοσης (MTU). Η ευθύνη για το μέγεθος της MTU βρίσκεται στο υψηλότερο πρωτόκολλο στρώματος. Το L2CAP χωρίζει το πακέτο σε μονάδες δεδομένων πρωτοκόλλου (PDU) που στέλνουν στο χαμηλότερα στρώματα. Συνήθως το L2CAP τρέχει επάνω από HCI όπου τα PDUs μετατρέπονται σε μονάδες baseband.
- ✦ Επανασυναρμολόγηση: Το πρωτόκολλο baseband σιγουρεύει ότι τα πακέτα είναι στη σειρά και ότι προστατεύεται η ακεραιότητα τους. Το ARQ χρησιμοποιείται, με τη βοήθεια και του CRC, για να διατηρηθεί η αξιοπιστία στην σύνδεση. Καθώς ο ελεγκτής baseband λαμβάνει τα πακέτα ACL αναφέρει στο στρώμα L2CAP την άφιξη κάθε πακέτου ή αποθηκεύει διάφορα πακέτα πριν τα αναφέρει.
- ✦ Ποιότητα της υπηρεσίας: Κάθε εφαρμογή L2CAP πρέπει να ελέγχει τους πόρους που χρησιμοποιούνται από το πρωτόκολλο και να εξασφαλίσει ότι το QoS διατηρείται στα επιθυμητά επίπεδα.
- ✦ Ομάδες: Πολλά πρωτόκολλα χρησιμοποιούν την έννοια μιας ομάδας διευθύνσεων. Το πρωτόκολλο baseband υποστηρίζει την έννοια ενός piconet. Η αφαίρεση ομάδας L2CAP είναι πιθανό να απεικονίσει αποτελεσματικά τις ομάδες πρωτοκόλλου προς τα piconets. Με την αφαίρεση ομάδας, τα υψηλότερου επιπέδου πρωτόκολλα δεν είναι απαραίτητο να μιλήσουν άμεσα με το baseband protocol και τον link manager. Το στοιχείο που στέλνεται σε ένα κανάλι ομάδας στέλνεται με έναν best-effort τρόπο. Τα κανάλια της ομάδας είναι αναξιόπιστα και δεν έχουν κανένα QoS. Το L2CAP δεν μπορεί να εγγυηθεί ότι τα δεδομένα φθάνουν σε όλα τα μέλη.
- ✦ Κανάλια: Ένα κανάλι αντιπροσωπεύει μια ροή δεδομένων μεταξύ των οντοτήτων L2CAP. Το προσδιοριστικό των καναλιών (CID) αναφέρεται σε κάθε endpoint του καναλιού L2CAP. Ένα CID είναι σχετικό με μια

ιδιαίτερη συσκευή και μια συσκευή μπορεί να ορίσει CIDs ανεξάρτητα από άλλες συσκευές.

2.6.3 Γεγονότα και ενέργειες

Γεγονότα:

Υπάρχουν πέντε κατηγορίες γεγονότων (εισερχόμενα μηνύματα στο στρώμα L2CAP):

1. Οι ενδείξεις και οι επιβεβαιώσεις από τα χαμηλότερα στρώματα.
2. Αιτήματα και απαντήσεις από τα υψηλότερα στρώματα.
3. Δεδομένα από τους peers (ομάδες).
4. Αιτήματα και απαντήσεις σημάτων από τους peers.
5. Γεγονότα που προκαλούνται από την λήξη των χρονομέτρων (timer expirations).

Ενέργειες:

Υπάρχουν πέντε κατηγορίες ενεργειών (εξερχόμενα μηνύματα από το στρώμα L2CAP):

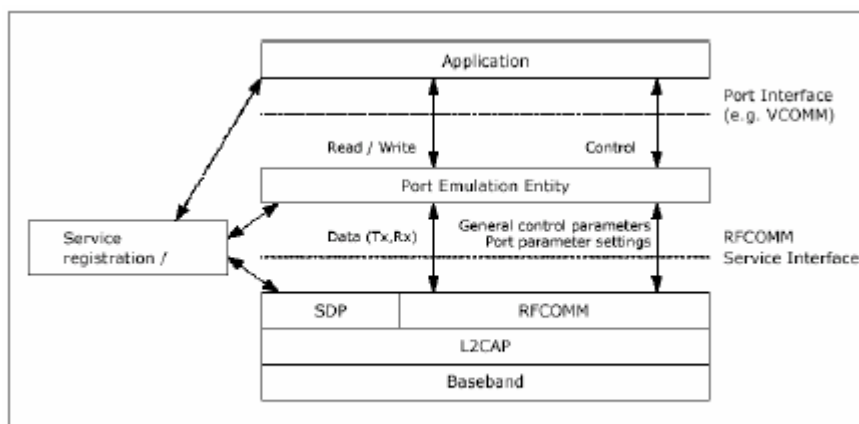
1. Επιβεβαιώσεις και ενδείξεις προς τα υψηλότερα στρώματα.
2. Αιτήματα και απαντήσεις προς τα χαμηλότερα στρώματα.
3. Αιτήματα και απαντήσεις προς τους peers.
4. Μετάδοση δεδομένων προς τους peers.
5. Έναρξη των χρονομέτρων.

Σηματοδότηση:

Το L2CAP πρέπει να είναι σε θέση να καθορίσει τη διεύθυνση Bluetooth της συσκευής στην οποία στέλνει τις εντολές. Πολλές εντολές μαζί μπορούν να σταλούν σε ένα ενιαίο πακέτο L2CAP.

2.7 RFCOMM

Το RFCOMM προορίζεται ως πρωτόκολλο αντικατάστασης καλωδίων και χρησιμοποιείται κυρίως σε εφαρμογές με άλλες γειτονικές συσκευές που έχουν σειριακές θύρες. Παρέχει τις υπηρεσίες του στα υψηλότερα στρώματα π.χ. στο πρωτόκολλο σημείου προς σημείο (PPP), στο πρωτόκολλο ανταλλαγής αντικειμένου (OBEX) και στα AT Commands. Είναι ένα απλό πρωτόκολλο μεταφορών που παρέχει transparent data stream και κανάλια ελέγχου πάνω από τα κανάλια L2CAP. Επιπλέον μπορεί να μιμηθεί την σειριακή θύρα RS232.



Μοντέλο αναφοράς RFCOMM

ΕΙΚΟΝΑ 2.7

Το στρώμα RFCOMM χρησιμοποιεί το L2CAP για να καθιερώσει τα κανάλια που χρησιμοποιούν οι συσκευές. Επομένως το L2CAP πρέπει να παρέχει κανάλια μεγάλης αξιοπιστίας για να εξασφαλίζει ότι τα πλαίσια παραλαμβάνονται όπως στάλθηκαν και ότι δεν υπάρχει κανένα αντίγραφο.

Το port emulation entity αποτελεί μια συγκεκριμένη διεπαφή επικοινωνίας συστημάτων (API) στις υπηρεσίες RFCOMM. Στα περισσότερα συστήματα το port emulation entity και το RFCOMM αποτελούν έναν port driver.

Η εγγραφή των μεμονωμένων εφαρμογών ή των υπηρεσιών, μαζί με τις πληροφορίες που πρέπει να φθάσουν σε αυτές (π.χ. το κανάλι υπηρεσιών RFCOMM) αποτελεί ευθύνη της εκάστοτε εφαρμογής.

2.7.1 Χαρακτηριστικά γνωρίσματα

Το RFCOMM είναι χτισμένο πάνω στο null modem scheme. Δύο συσκευές Bluetooth μπορούν να ανοίξουν πολλαπλές emulated σειριακές θύρες μεταξύ τους. Συγκεκριμένα το RFCOMM υποστηρίζει μέχρι και 60 ανοιχτές emulated θύρες. Το πόσες από αυτές μπορούν να χρησιμοποιηθούν καθορίζεται από την εφαρμογή.

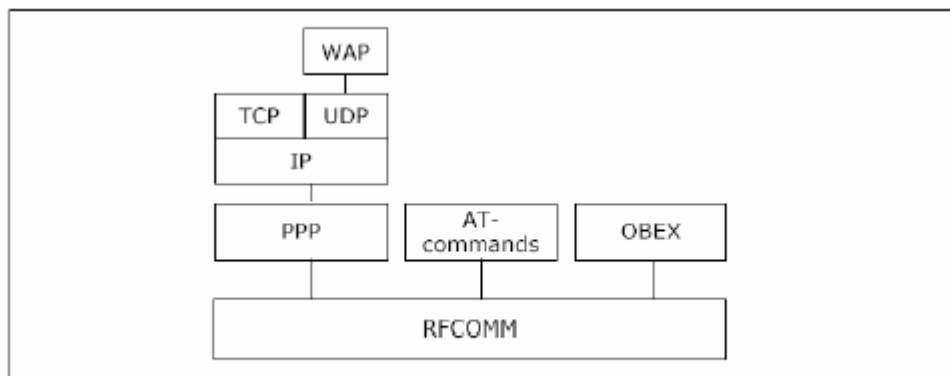
Εάν όλα τα κανάλια προς μια ορισμένη συσκευή Bluetooth δεν απασχολούνται για έναν προκαθορισμένο χρόνο, το RFCOMM επιτρέπει σε εκείνη την συσκευή να μπει σε κατάσταση εξοικονόμησης ισχύος χωρίς οποιαδήποτε παρέμβαση.

2.7.2 Υψηλότερα στρώματα

Το RFCOMM καθιστά την σειριακή θύρα μηχανισμό μεταφοράς και επιτρέπει την χρήση της από τα υψηλότερα πρωτόκολλα. Τρία είδη υψηλότερων πρωτοκόλλων μπορούν να διασυνδεθούν με το RFCOMM.

Το από σημείο σε σημείο πρωτόκολλο (PPP), που έχει σαν σκοπό να πάρει τα πακέτα IP από/προς το στρώμα PPP και να τα τοποθετήσει στο τοπικό LAN.

Η εφαρμογή αυτού του πρωτοκόλλου στις συσκευές Bluetooth, κάνει πιθανή την επικοινωνία με οποιαδήποτε άλλη συσκευή που συνδέεται με το Internet. Το TCP/IP/PPP χρησιμοποιείται για όλα τα σενάρια χρήσης γεφυρών Internet ενώ το UDP/IP/PPP για μεταφορά μέσω των ασύρματων πρωτοκόλλων εφαρμογής (WAP).



Τα υψηλότερα επίπεδα που σχετίζονται με το RFCOMM επίπεδο

ΕΙΚΟΝΑ 2.7.2

Τα AT commands καθιστούν δυνατή την ενσωμάτωση και τον έλεγχο των υπηρεσιών FAX, των modems και των κινητών τηλεφώνων. Διάφορα AT commands έχουν καθοριστεί από το Bluetooth SIG για να ελέγξουν αυτές τις υπηρεσίες. Τα AT command είναι βασισμένα στο ITU-T, στη σύσταση B.250 και στο ETS 300.916 (GSM 07.07).

Το OBEX περιγράφεται ως πολύ ελαφρύτερη έκδοση του HTTP και είναι ένα πρωτόκολλο συνόδου που αναπτύσσεται από την Infrared Data Association. Ο κύριος στόχος του είναι η ανταλλάγη αντικειμένων (browsing, δημιουργία, αντιγραφή κ.λ.π) με έναν απλό τρόπο.

Πάνω από τα πρωτόκολλα που περιγράφονται ανωτέρω, μπορούν να χτιστούν εφαρμογές και για υψηλότερα πρωτόκολλα.

2.7.3 Συμπέρασμα

Από τους κατασκευαστές των τυποποιημένων μη-ασύρματων συσκευών points of views, που επιθυμούν να καταστήσουν τα προϊόντα τους ασύρματα, το Bluetooth και ειδικά το RFCOMM είναι ένας ελκυστικός τρόπος να παρασχεθεί μια γρήγορη και απλή λύση. Το υπόλοιπο πρόβλημα είναι η παροχή ηλεκτρικού ρεύματος. Επιπλέον οι κατασκευαστές των προϊόντων WAP μπορούν εύκολα να ενσωματώσουν το Bluetooth στα προϊόντα τους χρησιμοποιώντας το RFCOMM για να υποστηρίξουν τα υψηλότερα στρώματα των πρωτοκόλλων.

2.8 Πρωτόκολλο ανακάλυψης υπηρεσιών (*Service Discovery Protocol*)

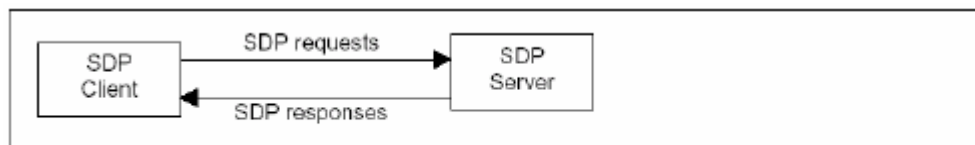
Το πρωτόκολλο ανακάλυψης υπηρεσιών (SDP) έχει σαν σκοπό να παρέχει στα Bluetooth entities τα μέσα, για να γνωρίζουν ποιες υπηρεσίες είναι διαθέσιμες. Είναι επίσης δυνατό να καθοριστούν τα χαρακτηριστικά αυτών των υπηρεσιών με τη χρησιμοποίηση του SDP. Το πρωτόκολλο πρέπει να είναι σε θέση να καθορίζει τις ιδιότητες οποιασδήποτε υπηρεσίας, τωρινής ή μελλοντικής, οποιασδήποτε πολυπλοκότητας και σε οποιοδήποτε λειτουργικό περιβάλλον.

Αυτό είναι ένα πολύ σημαντικό ζήτημα για το Bluetooth, δεδομένου ότι το μέγεθος και το πλήθος των υπηρεσιών αναμένεται να αυξηθεί γρήγορα και ίσως κατά τρόπο ανεξέλεγκτο.

Το SDP μπορεί να χρησιμοποιηθεί μόνο για την έρευνα των υπηρεσιών και τη συλλογή των πληροφοριών, π.χ. πρόσβαση στις ιδιότητες και πρόσβαση στα σχετικά πρωτοκόλλα υπηρεσιών τους. Έχει χαμηλή πολυπλοκότητα και χαμηλές απαιτήσεις για τα χαμηλότερα στρώματα. Στην πραγματικότητα μπορεί να λειτουργήσει ακόμη και με μια αναξιόπιστη μεταφορά.

2.8.1 Client – server

Δύο συσκευές Bluetooth λαμβάνουν μέρος σε μια SDP συναλλαγή, ο SDP server και ο SDP client. Ο server παρέχει τις υπηρεσίες που ο client ανακαλύπτει και εξετάζει, καθώς έχει έναν κατάλογο στον οποίο περιγράφονται τις υπηρεσίες του. Τα στοιχεία του καταλόγου καλούνται εγγραφές υπηρεσιών.



Απλοποιημένο μοντέλο Client-Server

EIKONA 2.8.1

Κάθε εγγραφή υπηρεσιών περιγράφει τα χαρακτηριστικά μιας υπηρεσίας. Ο client ανακτά τις πληροφορίες από μία εγγραφή υπηρεσιών με την αποστολή ενός SDP αιτήματος στον server. Ο server μπορεί να ανταποκριθεί στα αιτήματα σχετικά με όλες τις υπηρεσίες που παρέχονται προς τους clients. Παράλληλα μόνο ένας SDP server ανά συσκευή Bluetooth απαιτείται.

Εντούτοις, μια συσκευή που ενεργεί μόνο ως client δεν χρειάζεται κανέναν SDP server. Ο SDP client μπορεί να χειριστεί τις πολλαπλές εφαρμογές clients σε παρόμοιο τρόπο. Ένας SDP server δεν μπορεί να ενημερώσει τους νέους clients για την παρουσία του, ούτε μπορεί να ειδοποιήσει τους clients πότε γίνεται μη διαθέσιμος, μέσω SDP. Αυτές οι καταστάσεις αντιμετωπίζονται από άλλα πρωτόκολλα στο Bluetooth Protocol Stack.

2.8.2 Εγγραφές υπηρεσιών και ιδιότητες υπηρεσιών

Όπως αναφέραμε και παραπάνω, ο SDP server ενημερώνει και αποθηκεύει έναν κατάλογο. Ο κατάλογος αυτός αποτελείται από τις εγγραφές υπηρεσιών, οι οποίες περιέχουν όλες τις πληροφορίες για τις υπηρεσίες. Οι εγγραφές υπηρεσιών αποτελούνται από έναν κατάλογο ιδιοτήτων υπηρεσιών. Κάθε αρχείο υπηρεσιών έχει μια (τοπικά) μοναδική υπηρεσία λήψης αρχείων που προσδιορίζει την εγγραφή.

Το SDP δεν έχει την ικανότητα να αφήσει έναν SDP server να ειδοποιήσει τους δικούς του clients για τις προστιθέμενες ή αφαιρούμενες εγγραφές υπηρεσιών. Επομένως, είναι σημαντικό ο SDP server να μην ανακυκλώνει την διαχείριση των εγγραφών υπηρεσιών ενώ μια σύνδεση L2CAP παραμένει καθιερωμένη.

Κάθε ιδιότητα υπηρεσιών αποτελείται από ένα attribute ID και μια attribute value. Μια attribute value μπορεί να περιέχει γενικές πληροφορίες σε αφηρημένο επίπεδο σχετικά με την πολυπλοκότητα, για να υποστηρίξει ένα ευρύ φάσμα υπηρεσιών και λειτουργικών περιβαλλόντων. Η attribute ID προσδιορίζει κάθε ιδιότητα υπηρεσιών σε μία εγγραφή υπηρεσιών. Οι attributes IDs και η συμπεριφορά των τιμών που συνδέονται με αυτές καθορίζονται στον καθορισμό κατηγορίας υπηρεσιών. Μια συγκεκριμένη ID έχει την ίδια σημασία σε όλες τις περιπτώσεις μιας κατηγορίας υπηρεσιών. Η έννοια της αξίας ιδιοτήτων καθορίζεται από την ταυτότητα ιδιοτήτων και επομένως από την κατηγορία υπηρεσιών της εγγραφής Value's Service.

2.8.3 Κατηγορίες υπηρεσιών (Service Classes)

Μια υπηρεσία είναι πάντα μια περίπτωση μιας κατηγορίας υπηρεσιών. Μια κατηγορία υπηρεσιών μπορεί να κληρονομήσει μια άλλη κατηγορία υπηρεσιών, που δημιουργεί μια sub-/super-class σχέση. Οι ομοιότητες με τις αντικειμενοστραφής γλώσσες προγραμματισμού είναι προφανείς: Μπορούμε να θεωρήσουμε την κατηγορία υπηρεσιών ως κλάση και την υπηρεσία ως

αντικείμενο, δηλαδή μια subclass είναι μια εκτεταμένη έκδοση της super class. Έχει την πλήρη λειτουργία της super-class και μιας ή περισσότερων πρόσθετων ιδιοτήτων.

2.8.4 Ψάχνοντας και κάνοντας browsing για τις υπηρεσίες

Ένας SDP client μπορεί να ψάξει για τις υπηρεσίες ανακτώντας το Service Record Handle για οποιοδήποτε εγγραφή υπηρεσιών. Η αναζήτηση είναι βασισμένη στις τιμές των ιδιοτήτων των εγγραφών υπηρεσιών. Ο SDP client μπορεί να ψάξει μόνο για τις ιδιότητες των οποίων οι τιμές είναι UUIDs (καθολικό μοναδικό προσδιοριστικό). Έτσι μια υπηρεσία δεν μπορεί να βρεθεί, εκτός αν τουλάχιστον μια από τις ιδιότητές της είναι UUID.

Ένα UUID είναι μια τιμή 128 bits που είναι εγγυημένα μοναδική. Τα UUIDs που θεωρούνται ότι χρησιμοποιούνται συχνά έχουν 32 ή 16 bit alias. Το αρχικό UUID μπορεί να υπολογιστεί με τη χρησιμοποίηση των τύπων που διευκρινίζονται στο Core Specification.

Όταν ο client αρχίζει τη συναλλαγή έρευνάς του, δημιουργεί ένα σχέδιο αναζήτησης υπηρεσιών, το οποίο είναι ένας κατάλογος ιδιοτήτων επιθυμητών υπηρεσιών. Είναι σαφές ότι αυτός ο κατάλογος περιέχει μόνο τα UUIDs. Το σχέδιο αναζήτησης υπηρεσιών ταιριάζει με μία εγγραφή υπηρεσίας εάν όλα τα στοιχεία του σχεδίου αναζήτησης υπηρεσιών ταιριάζουν με οποιεσδήποτε από τις ιδιότητες των εγγραφών υπηρεσίας.

Εάν ένας SDP client επιθυμεί να ερευνήσει ποιες υπηρεσίες είναι διαθέσιμες, μπορεί να χρησιμοποιήσει μια διαδικασία που αποκαλείται browsing. Για να επιτραπεί το ξεφύλλισμα, όλες οι κατηγορίες υπηρεσιών περιέχουν μια ιδιότητα αποκαλούμενη BrowseGroupList η τιμή της οποίας είναι ένας κατάλογος που ξεφυλλίζει τις ομάδες. Οι υπηρεσίες του server συνδέονται με μία από αυτές τις browse ομάδες.

2.8.5 Συμπέρασμα

Το πρωτόκολλο ανακάλυψης υπηρεσιών (Service Discovery Protocol) είναι ένα ενδιαφέρον μέρος της προδιαγραφής Bluetooth. Το γεγονός ότι έχει μικρό μέγεθος, μπορεί να φανεί παράξενο, δεδομένου ότι οι ικανότητές του είναι απεριόριστες. Οι υπηρεσίες οποιασδήποτε φύσης και σε οποιοδήποτε λειτουργικό περιβάλλον μπορούν να ανακαλυφθούν. Κατά κάποιο τρόπο, το SDP είναι το πρωτόκολλο που είναι αρμόδιο για την επιτυχία του Bluetooth: Με το SDP που λειτουργεί όπως περιγράφεται στην προδιαγραφή έχουμε την δυνατότητα της διασύνδεσης όλου του ηλεκτρονικού εξοπλισμού κατά τρόπο φιλικό προς το χρήστη. Αφ' ετέρου, εάν το SDP δεν λειτουργεί όπως θα έπρεπε ή εάν δεν χρησιμοποιείται κατάλληλα, θα εμφανιστούν προβλήματα. Ο εξοπλισμός Bluetooth θα διαμορφώσει form piconets, αλλά είναι σχεδόν απίθανο η λειτουργία σε επίπεδο χρήστη να είναι όπως επιδιώκεται.

Το SDP είναι το πρωτόκολλο που δίνει στο Bluetooth κάτι περισσότερο από ότι τα πρότυπα αντικατάστασης καλωδίων. Μας δίνει την ευκαιρία να κατασκευάσουμε τον ηλεκτρονικό εξοπλισμό ώστε να βασίζεται στις εξ ολοκλήρου νέες λειτουργίες και να μπορεί να βρει εφαρμογή σε πολύ ευρύτερο πεδίο.

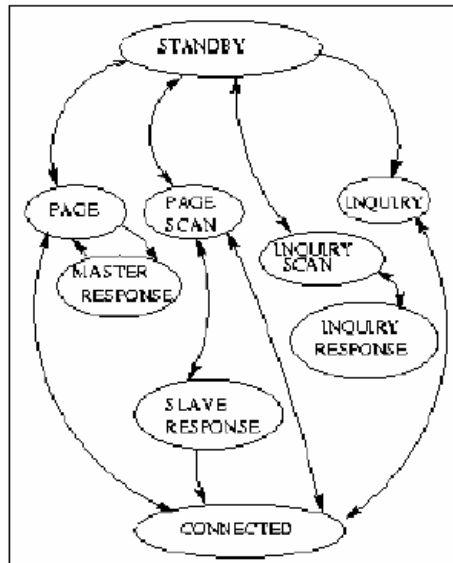
2.9 Εγκατάσταση Συνδέσεων

Η εγκατάσταση συνδέσεων στο Bluetooth όπως ήδη αναφέραμε, έχει να κάνει με τη δημιουργία *piconets*. Η όλη διαδικασία για τη δημιουργία ενός *piconet* μεταξύ δύο συσκευών – κόμβων μπορεί να περιγραφεί από μια εναλλαγή καταστάσεων στις οποίες περνάνε ακολουθιακά οι κόμβοι. Οι καταστάσεις αυτές φαίνονται στο σχήμα της επόμενης σελίδας και θα συνεχίσουμε δίνοντας μια σύντομη περιγραφή για κάθε μια από αυτές.

Κατ' αρχήν, κάθε συσκευή έχει ως *default* κατάσταση την κατάσταση **Standby**. Αυτή είναι μια κατάσταση ελάχιστης κατανάλωσης ενέργειας. Οι καταστάσεις στις οποίες μπορεί να μεταπηδήσει ένας κόμβος τότε, είναι οι **Inquiry**, **Inquiry Scan**, **Page**, ή **Page Scan**.

Inquiry

Σε αυτή την κατάσταση, η συσκευή προσπαθεί να εντοπίσει άλλες συσκευές που βρίσκονται στον περιβάλλοντα χώρο, μεταδίδοντας ένα κατάλληλο πακέτο σε συγκεκριμένες συχνότητες. Αν αυτή η διαδικασία γίνεται αυτόματα θα πρέπει να έχει ληφθεί μέριμνα ώστε να αποφευχθεί ο συγχρονισμός και κατ' επέκταση οι συγκρούσεις με τα αντίστοιχα πακέτα μια άλλης συσκευής.



ΕΙΚΟΝΑ 2.9

Inquiry Scan

Σε αυτή την κατάσταση, οι συσκευές που επιτρέπουν την ανακάλυψη τους από τις υπόλοιπες, εισέρχονται περιοδικά, και μπορούν να ακούσουν για μηνύματα **Inquiry** στις αντίστοιχες συχνότητες. Οι συσκευές μπορούν να εισέλθουν σε αυτή την κατάσταση από την κατάσταση **Inquiry** ή από την κατάσταση **Connected**. Αν η είσοδος γίνεται από την τελευταία, οι ζεύξεις SCO παραμένουν, ενώ οι ACL διακόπτονται. Η παρουσία ζεύξεων SCO μπορεί να παρατείνει τη διάρκεια της διαδικασίας **Inquiry**.

Inquiry Response

Όταν ληφθεί ένα μήνυμα **Inquiry**, αποστέλλεται πίσω ένα μήνυμα που περιέχει τη διεύθυνση του κόμβου και στοιχεία του χρονισμού του. Τα στοιχεία αυτά αποθηκεύονται από τον κόμβο που εκκίνησε τη διαδικασία και θα χρησιμοποιηθούν όταν απαιτηθεί κάποια σύνδεση μεταξύ αυτών των δύο

κόμβων. Μετά το τέλος αυτής της αρχικής διαδικασίας ο κάθε κόμβος έχει πληροφορίες για τις διευθύνσεις των γειτόνων του και όταν απαιτηθεί θα ξεκινήσει τη διαδικασία **Page**.

Page

Όπως είπαμε, ένας κόμβος που θα θελήσει να δημιουργήσει μια σύνδεση – αναλαμβάνει έτσι το ρόλο του *master* – θα περιέλθει στην κατάσταση **Page**. Ο *master*, θα αποστείλει ένα κατάλληλο μήνυμα σε προκαθορισμένες συχνότητες, ζητώντας να επικοινωνήσει με τον συγκεκριμένο κόμβο που επιθυμεί, του οποίου βέβαια γνωρίζει τη διεύθυνση. Η διαδικασία αυτή διαρκεί έως ότου ληφθεί το μήνυμα **Page Response**, όπως θα δούμε παρακάτω.

Page Scan

Στην κατάσταση αυτή εισέρχονται περιοδικά οι συσκευές από τις καταστάσεις **Standby** και **Connected**. Ο *slave* ακούει στις κατάλληλες συχνότητες για πακέτα **Page** που φέρουν τη δική του διεύθυνση.

Page Response

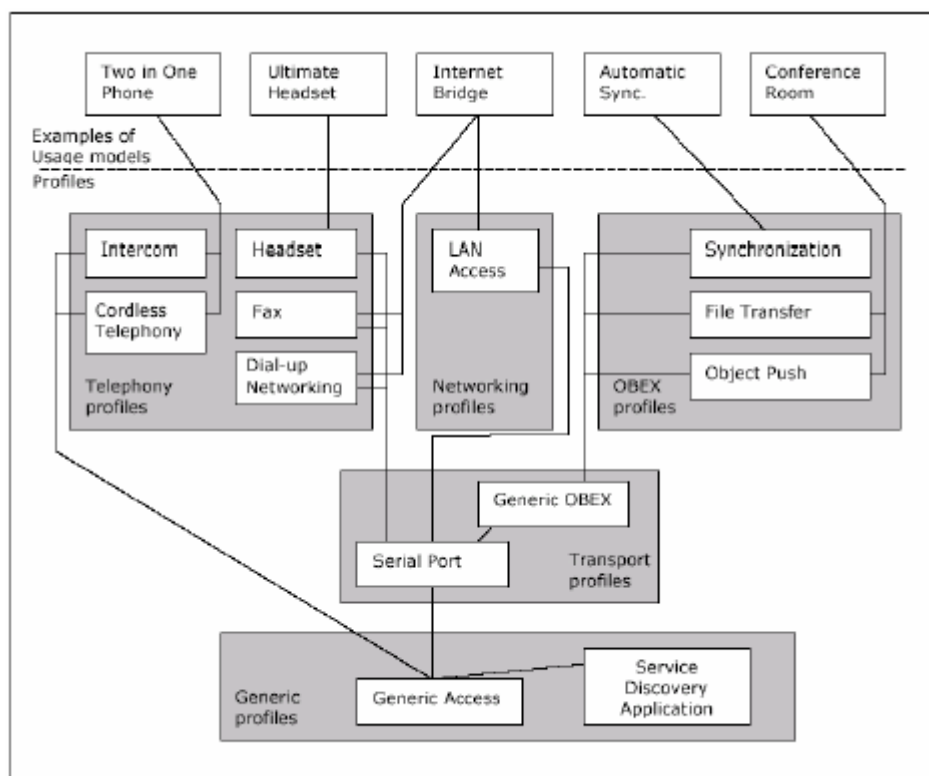
Μετά τη λήψη ενός **Page** μηνύματος, ο *slave* εισέρχεται σε αυτή την κατάσταση. Σε περίπτωση αποδοχής της αίτησης του *master* για επικοινωνία, ο *slave* θα πρέπει να αποστείλει ένα κατάλληλο μήνυμα αποδοχής ώστε να περάσει και ο *master* στην ίδια κατάσταση. Τότε, ο *master* αποστέλλει ένα πακέτο το οποίο θα περιλαμβάνει πληροφορίες για το χρονισμό του και τη διεύθυνσή του, ώστε να μπορέσει ο *slave* να προσδιορίσει την ακολουθία των αλμάτων στη συχνότητα στην οποία θα πραγματοποιείται από εκείνη τη στιγμή και πέρα η επικοινωνία.

Μετά την επιτυχή ολοκλήρωση της όλης διαδικασίας, ουσιαστικά έχει δημιουργηθεί ένα *piconet* μεταξύ του *master* και του *slave* και οι συσκευές περνούν στην κατάσταση **Connected**, οπότε μπορούν να εγκαταστήσουν τις απαραίτητες για την επικοινωνία ζεύξεις.

3 BLUETOOTH PROFILES

Τα διάφορα bluetooth profiles αναπτύχθηκαν με σκοπό να περιγράψουν πως γίνεται η υλοποίηση των διαφόρων προτύπων χρήσης (usage models). Τα πρότυπα χρήσης με την σειρά τους περιγράφουν τα βασικά σενάρια εφαρμογών για συσκευές Bluetooth, όπου η επικοινωνία γίνεται με την βοήθεια της τεχνολογίας Bluetooth. Η έννοια του προφίλ είναι πολύ σημαντική, καθώς με την βοήθεια του επιτυγχάνεται η διαλειτουργικότητα. Για να εξασφαλιστεί η επιτυχής επικοινωνία μεταξύ των συσκευών Bluetooth, απαιτείται η συνεργασία τριών κύριων επιπέδων. Με το επίπεδο radio, εξασφαλίζεται ότι οι συσκευές μπορούν να έρθουν σε επαφή. Με το επίπεδο πρωτοκόλλου, εξασφαλίζεται ότι οι συσκευές μπορούν να “μιλήσουν”, ενώ με τα μοντέλα χρήσης, εξασφαλίζεται ότι οι συσκευές μπορούν να συνεργαστούν για την εκτέλεση των διάφορων εφαρμογών του τελικού χρήστη.

Τα προφίλ Bluetooth είναι βασισμένα στο πρότυπο διασύνδεσης ανοικτών συστημάτων (OSI). Κάθε προφίλ είναι βασισμένο σε ορισμένα θεμελιώδη προφίλ Bluetooth και συσχετίζεται με ένα ή περισσότερα usage models. Στην προδιαγραφή καθορίζονται πέντε ομάδες προφίλ: Προφίλ τηλεφωνίας (Telephony Profiles), προφίλ δικτύωσης (Networking Profiles), προφίλ OBEX (OBEX Profiles), προφίλ μεταφορών (Transport Profiles) και γενικά προφίλ (Generic Profiles):



Εικόνα 3 Το γενεολογικό δέντρο των profile και η σχέση usage models - profiles

3.1 Γενικά προφίλ

Το γενικό προφίλ πρόσβασης (Generic Access Profile - GAP) και το προφίλ ανακάλυψης υπηρεσιών εφαρμογών (Service Discovery Application Profile - SDAP), αποτελούν τα δύο σημαντικότερα γενικά προφίλ.

Το GAP ορίζει τις γενικές διαδικασίες οι οποίες σχετίζονται με την ανακάλυψη συσκευών Bluetooth, καθώς και με θέματα του LMP για την σύνδεση σε μία συσκευή. Ο κύριος σκοπός του GAP είναι να περιγράψει πώς χρησιμοποιούνται τα χαμηλότερα στρώματα (LC και LMP). Ασχολείται επίσης με ζητήματα σχετικά με την ασφάλεια. Τα υπόλοιπα προφίλ κληρονομούν από το GAP χρησιμοποιώντας τις γενικές διαδικασίες που παρέχει.

Οι κυριότερες λειτουργίες του GAP είναι:

- Έλεγχος των απαιτήσεων διασύνδεσης με τον χρήστη (σωστά ονόματα, τιμές και σχέδια κωδικοποίησης).
- Modes (ανακαλυψιμότητα, σύνδεση δυνατότητας, ικανοί τρόποι ζεύξης).
- Πτυχές ασφάλειας (επικύρωση).
- Διαδικασίες αδρανούς κατάστασης (έρευνα, ανακάλυψη ονόματος, ζεύξη κ.λ.π).
- Διαδικασίες καθιέρωσης (περιγραφή γενικών διαδικασιών που μπορούν να χρησιμοποιηθούν).

Το SDAP καθορίζει τις διαδικασίες και τα πρωτόκολλα (κυρίως το SDP) που πρέπει να χρησιμοποιήσει μια εφαρμογή της bluetooth συσκευής, για να ανακαλύψει τις υπηρεσίες, καθώς και συγκεκριμένες πληροφορίες σχετικά με υπηρεσίες που υπάρχουν σε άλλες συσκευές.

3.2 Προφίλ μεταφορών (*Transport Profiles*)

Τα προφίλ μεταφορών περιέχουν το προφίλ σειριακής θύρας και τα γενικά προφίλ ανταλλαγής αντικειμένου (OBEX).

Το προφίλ σειριακής θύρας περιγράφει και καθορίζει τα πρωτόκολλα (κυρίως RFCOMM) και τις διαδικασίες που χρησιμοποιούνται από τις συσκευές που χρησιμοποιούν Bluetooth για να προσομοιώσουν την σειριακή επικοινωνία RS232. Η εφαρμογή και στις δύο πλευρές αποτελεί χαρακτηριστική εφαρμογή legacy. Οι εφαρμογές legacy θέλουν να επικοινωνήσουν μέσω σειριακού καλωδίου. Το καλώδιο προσωμοιώνεται αλλά οι εφαρμογές legacy δεν ξέρουν τίποτα για τις διαδικασίες Bluetooth για setting up emulated σειριακά καλώδια. Χρειάζονται όμως τη βοήθεια από κάποια είδους Bluetooth-aware helper εφαρμογή και στις δύο πλευρές, του προφίλ σειριακής θύρας.

Το γενικό προφίλ ανταλλαγής αντικειμένου (OBEX) καθορίζει τα πρωτόκολλα και τις διαδικασίες που θα χρησιμοποιηθούν από τις εφαρμογές, οι οποίες βασίζονται σε μοντέλα χρήσης τα οποία χρειάζονται την ανταλλαγή

αντικειμένων. Το μοντέλο χρήσης μπορεί να είναι, παραδείγματος χάριν, συγχρονισμός, μεταφορά αρχείων, ή ώθηση αντικειμένου (Object Push).

3.2.1 Προφίλ ώθησης αντικειμένου

Το προφίλ ώθησης αντικειμένου καθορίζει τις απαιτήσεις για τα πρωτόκολλα και τις διαδικασίες που θα χρησιμοποιηθούν από τις εφαρμογές που παρέχουν το μοντέλο χρήσης “ώθηση αντικειμένου”. Αυτό το προφίλ χρησιμοποιεί το γενικό προφίλ ανταλλαγής αντικειμένου (GOEP) για να καθορίσει τις απαιτήσεις παράλληλης λειτουργίας για τα πρωτόκολλα που απαιτούνται από τις εφαρμογές. Οι πιο κοινές συσκευές που χρησιμοποιούν αυτά τα πρότυπα χρήσης μπορεί να είναι notebook PCs, PDAs, και κινητά τηλέφωνα.

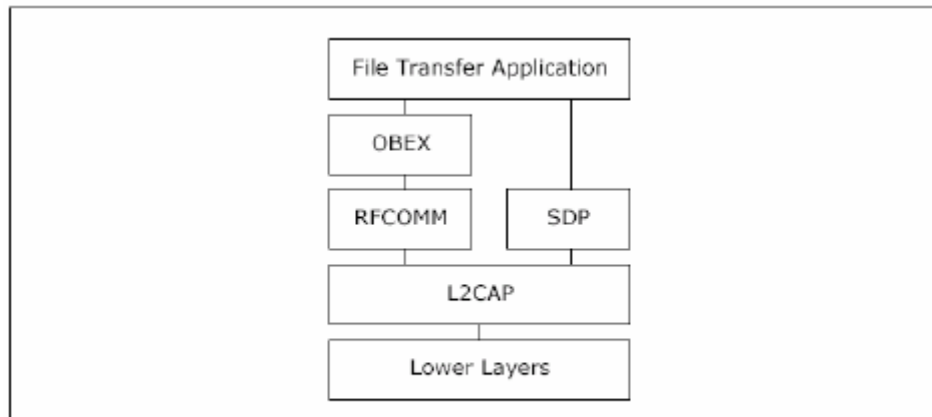
Τα σενάρια που καλύπτονται από αυτό το προφίλ είναι τα ακόλουθα:

- ✦ Χρήση μιας Bluetooth συσκευής, π.χ. ένα κινητό τηλέφωνο για να ωθήσει ένα αντικείμενο στο inbox μιας άλλης συσκευής Bluetooth. Το αντικείμενο μπορεί παραδείγματος χάριν να είναι μια επιχειρησιακή κάρτα ή ένα ραντεβού.
- ✦ χρήση μιας συσκευής Bluetooth π.χ. ένα κινητό τηλέφωνο για να τραβήξει μια επιχειρησιακή κάρτα από μια άλλη συσκευή Bluetooth.
- ✦ Χρήση μιας συσκευής Bluetooth π.χ. ένα κινητό τηλέφωνο στις επιχειρησιακές κάρτες ανταλλαγής με μια άλλη συσκευή Bluetooth. Ανταλλαγή ορίζεται ως μια ώθηση μιας επιχειρησιακής κάρτας που ακολουθείται από ένα τράβηγμα μιας επιχειρησιακής κάρτας.

3.2.2 Προφίλ Μεταφοράς Αρχείων

Το προφίλ μεταφοράς αρχείων διαμορφώνεται από το γενικό προφίλ πρόσβασης και το γενικό προφίλ ανταλλαγής αντικειμένου. Οι κοινές συσκευές που χρησιμοποιούν αυτό το προφίλ μπορούν να είναι PCs, notebooks και PDAs. Υπάρχουν τρία κύρια σενάρια χρήσης:

1. Browsing και πλοήγηση του συστήματος αρχείων μιας ενός συσκευής Bluetooth. Π.χ. ένα PC που κάνει browsing την ιεραρχία φακέλων ενός PDA.
2. Φάκελοι και αρχεία μεταφοράς μεταξύ δύο συσκευών Bluetooth.
3. Διαγραφή και δημιουργία φακέλων σε μια άλλη συσκευή Bluetooth.



Protocol Stack για εφαρμογές μεταφοράς αρχείων

EIKONA 3.2.2

3.2.3 Προφίλ Συγχρονισμού

Το προφίλ συγχρονισμού ορίζει τις απαιτήσεις που πρέπει να πληρούν οι εφαρμογές που παρέχουν το σενάριο χρήσης “συγχρονισμός”. Τυπικό σενάριο είναι ο συγχρονισμός PIM δεδομένων ανάμεσα σε δύο Bluetooth συσκευές, όταν η μια βρεθεί μέσα στην εμβέλεια της άλλης.

3.3 Προφίλ τηλεφωνίας (*Telephony Profiles*)

Τα προφίλ τηλεφωνίας μπορούν να διακριθούν σε δύο ομάδες. Η μία ομάδα είναι βασισμένη στο προφίλ σειριακής θύρας (στις AT commands) και σε αυτήν ανήκουν το προφίλ Headset, το προφίλ fax και το Dial-up Networking Profile. Αυτά τα προφίλ μπορούν να χρησιμοποιηθούν ως ασύρματη εναλλακτική λύση κατά τη σύνδεση των συσκευών π.χ. ένα modem σε ένα δημόσιο

τηλεφωνικό δίκτυο. Η άλλη ομάδα είναι βασισμένο στον έλεγχο τηλεφωνίας (TCS) και μπορεί να χρησιμοποιηθεί σε μια walkie-talkie λύση Bluetooth ή στα τηλέφωνα κατοικιών. Το ασύρματο προφίλ τηλεφωνίας και το προφίλ ενδοσυνεννοήσεων καλύπτουν αυτά τα μοντέλα χρήσης.

3.4 Προφίλ δικτύωσης

Το προφίλ πρόσβασης στο τοπικό LAN χρησιμοποιείται κυρίως στους προσωπικούς υπολογιστές με το PPP πρωτόκολλο για την πρόσβαση στο τοπικό LAN μέσω ενός ασύρματου σημείου πρόσβασης στο δίκτυο. Το προφίλ περιγράφει τρεις τρόπους για το PPP:

1. Πρόσβαση στο τοπικό LAN για μια ενιαία συσκευή Bluetooth.
2. Πρόσβαση στο τοπικό LAN για τις πολλαπλές συσκευές Bluetooth.
3. PC σε PC που χρησιμοποιεί τη δικτύωση PPP μέσω της προσομοίωσης σειριακής θύρας.

Το προφίλ πρόσβασης στο τοπικό LAN χρησιμοποιεί το PPP πάνω από το RFCOMM.

3.5 Συμπέρασμα

Καθώς αναπτύσσονται νέες εφαρμογές Bluetooth, η υψηλή απόδοση και η έλλειψη λαθών, είναι ζητήματα μεγάλης σημασίας. Τα βασικά προφίλ Bluetooth αποτελούν τα θεμέλια για την ανάπτυξη ανώτερων και πιο εξειδικευμένων προφίλ. Κάθε συσκευή Bluetooth πρέπει να εξασφαλίσει τη διαλειτουργικότητα και τη συνύπαρξη της με άλλες και αυτό επιτυγχάνεται με την κληρονομική σχέση του γενικού προφίλ πρόσβασης με τα υπόλοιπα προφίλ. Η σχέση αυτή καθιστά επίσης δυνατή την επαναχρησιμοποίηση των πρωτοκόλλων. Δεδομένου ότι τα πρότυπα χρήσης αυξάνονται γρήγορα, η χρήση των προφίλ είναι επιβεβλημένη ανάγκη, καθώς καθιστά απεριόριστες τις δυνατότητες για τις μελλοντικές εφαρμογές.

4 Τι είναι τα LINUX;

4.1 Εισαγωγή

Στο κεφάλαιο αυτό θα προσπαθήσουμε να εξηγήσουμε τι είναι το Linux, πως δημιουργήθηκε και από ποιους και το πώς έγινε το λειτουργικό σύστημα που είναι σήμερα.

4.2 Ιστορία του Linux

4.2.1 UNIX

Προκειμένου να γίνει κατανοητή η δημοτικότητα Linux, πρέπει να ταξιδεψουμε πίσω, περίπου 30 έτη πριν...

Φανταστείτε τους υπολογιστές τόσο μεγάλους όσο τα σπίτια, ακόμη και τα στάδια. Όπως είναι φυσικό το μέγεθος αυτών, προκαλούσε πολλά προβλήματα. Πέρα όμως από το μέγεθος, υπήρχε και κάτι ακόμα που έκανε τα πράγματα ακόμα χειρότερα: κάθε υπολογιστής είχε ένα διαφορετικό λειτουργικό σύστημα. Το λογισμικό προσαρμοζόταν πάντα για να εξυπηρετεί έναν συγκεκριμένο σκοπό, και επίσης, το λογισμικό για ένα σύστημα δεν έτρεχε σε άλλο. Το να είσαι σε θέση να εργαστείς με ένα σύστημα αυτόματα δεν σήμανε ότι θα μπορούσες να εργαστείς και με άλλο. Αυτό έκανε τα πράγματα δύσκολα, και για τους χρήστες και για τους διοικητές συστημάτων.

Το κόστος των υπολογιστών τότε ήταν πολύ μεγάλο και το γενικό κόστος αυξανόταν ακόμα περισσότερο με την εκπαίδευση που χρειαζόντουσαν οι χειριστές των μηχανημάτων αυτών. Όπως γίνεται φανερό το κόστος ήταν τεράστιο.

Τεχνολογικά ο κόσμος δεν ήταν αρκετά προηγμένος, έτσι έπρεπε να ζήσουν με αυτό το μέγεθος των υπολογιστών για μια άλλη δεκαετία. Το 1969, μια

ομάδα υπεύθυνων για την ανάπτυξη εργαστηρίων κουδουνιών ξεκίνησε να εργάζεται για τη λύση του προβλήματος της συμβατότητας λογισμικού.

Έτσι αναπτύξανε ένα καινούργιο λειτουργικό σύστημα (Operating System) του οποίου τα χαρακτηριστικά ήταν:

- Απλό και «Κομψό»
- Γραμμένο σε γλώσσα C αντί για γλώσσα μηχανής,
- Ικανός να ανακυκλώσει τον κώδικα.

Η ομάδα αυτή ονόμασε το project τους “UNIX”. Τα χαρακτηριστικά γνωρίσματα ανακύκλωσης κώδικα ήταν πολύ σημαντικά. Μέχρι τότε, όλα τα εμπορικά διαθέσιμα συγκροτήματα ηλεκτρονικών υπολογιστών γράφτηκαν σε έναν κώδικα που αναπτυσσόταν συγκεκριμένα για ένα σύστημα. Το Unix αφ' ετέρου χρειάστηκε μόνο ένα μικρό κομμάτι εκείνου του ειδικού κώδικα, ο οποίος τώρα συνήθως ονομάζεται πυρήνας(Kernel). Αυτός ο πυρήνας είναι το μόνο κομμάτι του κώδικα που πρέπει να προσαρμοστεί σε ένα συγκεκριμένο σύστημα και είναι κι αυτό που διαμορφώνει τη βάση του συστήματος Unix. Λειτουργικό σύστημα και όλες οι άλλες λειτουργίες χτίστηκαν πάνω από αυτόν τον πυρήνα και γράφτηκαν σε μια υψηλότερη γλώσσα προγραμματισμού, τη γλώσσα C. Αυτή η γλώσσα αναπτύχθηκε ειδικά για τη δημιουργία του συστήματος Unix. Χρησιμοποιώντας αυτήν την νέα τεχνική, ήταν πολύ ευκολότερο να αναπτυχθεί ένα λειτουργικό σύστημα που θα μπορούσε να τρέξει σε πολλούς διαφορετικούς τύπους υλικού.

Οι προμηθευτές λογισμικού προσαρμόστηκαν γρήγορα, δεδομένου ότι μπόρεσαν να πωλήσουν δέκα φορές περισσότερο λογισμικό, σχεδόν αβίαστα. Παράξενες νέες καταστάσεις ήρθαν: φανταστείτε παραδείγματος χάριν υπολογιστές από διαφορετικούς προμηθευτές να επικοινωνούν στο ίδιο δίκτυο, ή χρήστες που δουλεύουν σε διαφορετικά συστήματα χωρίς την ανάγκη για επιπρόσθετη εκπαίδευση να χρησιμοποιούν έναν άλλο υπολογιστή. Το Unix βοήθησε πολύ τους χρήστες να εξοικειωθούν με διαφορετικά συστήματα. Τις επόμενες δύο δεκαετίες η ανάπτυξη του UNIX συνεχιζόταν. Όλο και περισσότερα πράγματα μπορούσαν να κάνουν και όλο και περισσότεροι προμηθευτές υλικού και λογισμικού έκαναν τα προϊόντα τους να υποστηρίζουν τα UNIX.

Το UNIX αρχικά μπορούσες να το συναντήσεις μόνο σε μεγάλους και κεντρικούς υπολογιστές. Για να έρθεις σε επαφή με ένα σύστημα UNIX έπρεπε να εργάζεσαι σε ένα πανεπιστήμιο, ή για την κυβέρνηση ή για μεγάλες οικονομικές εταιρίες. Αλλά οι μικρότεροι υπολογιστές αναπτύσσονταν με γοργούς ρυθμούς και μέχρι το τέλος της δεκαετίας του '80, πολλοί άνθρωποι είχαν αποκτήσει προσωπικούς υπολογιστές. Έως τότε, υπήρξαν διάφορες εκδόσεις του Unix διαθέσιμες για την αρχιτεκτονική των PC, αλλά καμία από αυτές δεν ήταν πραγματικά ελεύθερη.

4.2.2 Linus and Linux

Ο Linus Torvalds ο οποίος σπούδαζε πληροφορική στο πανεπιστήμιο του Helsinki σκέφτηκε πως θα ήταν καλή ιδέα να υπήρχε μια ελεύθερη ακαδημαϊκή έκδοση του UNIX και αμέσως χωρίς να χάνει χρόνο, ξεκίνησε να προγραμματίζει. Στη συνέχεια, άρχισε να υποβάλλει τις ερωτήσεις, ψάχνοντας τις απαντήσεις και τις λύσεις που θα τον βοηθούσαν να εγκαταστήσει το Unix στο PC του. Παρακάτω είναι μια από τα πρώτα post του σε comp.os.minix, που το έστειλε στις 3 Ιουλίου το 1991:

From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)

Newsgroups: comp.os.minix

Subject: Gcc-1.40 and a posix-question

Message-ID: <1991Jul3.100050.9886@klaava.Helsinki.FI>

Date: 3 Jul 91 10:00:50 GMT

Hello netlanders,

Due to a project I'm working on (in minix), I'm interested in the posix standard definition. Could somebody please point me to a (preferably) machine-readable format of the latest posix rules? Ftp-sites would be nice.

Από την πρώτη στιγμή ο στόχος του Torvald ήταν να δημιουργήσει ένα πλήρως ελεύθερο σύστημα το οποίο ήταν ουσιαστικά σαν το UNIX. Αυτός ήταν λόγος που στο παραπάνω post ζήτησε τα POSIX πρότυπα, που ακόμα είναι τα πρότυπα για το Unix.

Σε εκείνες τις μέρες οι plug and play συσκευές δεν είχα ανακαλυφθεί ακόμα, παρόλα αυτά όλο και περισσότεροι άνθρωποι εκφράζανε την επιθυμία να έχουν σύστημα UNIX στους υπολογιστές τους. Όλο και περισσότεροι drivers για συσκευές γινόντουσαν διαθέσιμοι και την πάροδο του χρόνου έτσι το παραπάνω πρόβλημα άρχισε γρήγορα να ξεπερνιέται. Επίσης υπήρχαν άνθρωποι οι οποίοι με το που γινόταν διαθέσιμο ένα υλικό, το δοκιμάζανε αμέσως στα LINUX να δουνε τη συμπεριφορά του και αμέσως φτιάχνανε software για αυτή τη συσκευή. Από τότε αυτοί οι άνθρωποι καλούνται ως "nerds" ή "freaks". Βέβαια όπως είναι φυσικό αυτούς καθόλου δεν τους ενοχλεί αυτή η ονομασία και ουσιαστικά είναι οι άνθρωποι οι οποίοι συντέλεσαν στο να μεγαλώνει όλο και περισσότερο η λίστα των υλικών που υποστηρίζουν τα LINUX.

Στη συνέχεια παρουσιάζουμε άλλο ένα post του Linus Torvalds σε comp.os.minix

From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)

Newsgroups: comp.os.minix

Subject: What would you like to see most in minix?

Summary: small poll for my new operating system

Message-ID: <1991Aug25.205708.9541@klaava.Helsinki.FI>

Date: 25 Aug 91 20:57:08 GMT

Organization: University of Helsinki

Hello everybody out there using minix -

I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. This has been brewing since april, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat (same physical layout of the file-system (due to practical reasons) among other things).

I've currently ported bash(1.08) and gcc(1.40), and things seem to work.

This implies that I'll get something practical within a few months, and I'd like to know what features most people would want. Any suggestions are welcome, but I won't promise I'll implement them :-)

Linus (torvalds@kruuna.helsinki.fi)

PS. Yes - it's free of any minix code, and it has a multi-threaded fs.

It is NOT protable (uses 386 task switching etc), and it probably never will support anything other than AT-harddisks, as that's all I have :-).

Πριν κλείσουμε αυτό το κεφάλαιο, θα δώσουμε κάποια στοιχεία για την γεωμετρική αύξηση των χρηστών LINUX με τη πάροδο του χρόνου και πόσο γρήγορα, «αγκαλιάσανε» την ιδέα του Torvald.

➤ **1991** Users: 1

Linus Torvalds, 21

Version:0.01

Size: 10,000 lines of code

➤ **1992** Users: 1,000

Λειτουργικό Σύστημα που «τρέχει» πλήρως σε τσιπάκια intel x86. Επίσης έχει προστεθεί ένα γραφικό περιβάλλον.

Version:0.96

Size: 40,000 lines of code

➤ **1993** Users: 20,000

Πάνω από 100 προγραμματιστές συμβάλλουν στις αλλαγές του κώδικα.

Version:0.99

Size: 100,000 lines of code

➤ **1994** Users: 100,000

Προστίθεται η ικανότητα δικτύωσης

Version: Linux 1.0

Size: 170,000 lines of code

➤ **1995** Users: 500,000

Τροποποιείται για να «τρέχει» σε Intel, Digital and Sun SPARC επεξεργαστές

Πουλήθηκαν πάνω από 10000 περιοδικά που αφορούσαν το LINUX

Version: Linux 1.2

Size: 250,000 lines of code

➤ **1996** Users: 1,500,000

Είχε την ικανότητα να χρησιμοποιεί τη δύναμη πολλών επεξεργαστών μαζί.

Version: Linux 2.0

Size: 400,000 lines of code

➤ **1997** Users: 3,500,000

Μηνιαία περιοδικά για το LINUX κυκλοφορούσαν σε Ιαπωνία, Πολωνία και Αμερική. Κάθε βδομάδα δημοσιευόταν και καινούργια έκδοση LINUX.

Version: Linux 2.1

Size: 800,000 lines of code

➤ **1998** Users: 7,500,000

Περίπου 10000 προγραμματιστές προσπαθούσαν για τη βελτίωση το LINUX.

Version: Linux 2.1.110 (110th update of 2.1)

Size: 1.5 million lines of code

➤ Συγκριτικά με τα Windows NT:

NT 3.51 - 6 million lines of code

NT 4.0 - 16 million lines of code

NT 5.0 - 27 million lines of code *so far*

4.3 Τι είναι τα LINUX;

Ουσιαστικά τα LINUX είναι ένα από τα πιο γνωστά λειτουργικά συστήματα, τα οποία κάνουν όλο ένα και πιο αισθητή την παρουσία τους στην αγορά και αυτή τη στιγμή είναι η μεγαλύτερη απειλή για τα WINDOWS και τη MICROSOFT. Τα LINUX όπως ήδη έχουμε αναφέρει είναι ένα Open Source λειτουργικό σύστημα.

4.3.1 Τι εννοούμε λέγοντας Open Source λειτουργικό σύστημα;

Καταρχήν λέγοντας Open Source λειτουργικό σύστημα εννοούμε ένα λειτουργικό σύστημα το οποίου ο κώδικας είναι «ανοιχτός» δηλαδή ο χρήστης μπορεί να επέμβει στην «καρδιά» του λειτουργικού συστήματος και να κάνει τις τροποποιήσεις που αυτός θέλει.

Η Φιλοσοφία που κρύβεται πίσω από την ιδέα ενός open source λειτουργικού συστήματος είναι αρκετά απλή και στηρίζεται στο ότι όταν οι προγραμματιστές μπορούν να διαβάζουν, να διανείμουν και να αλλάζουν τον κώδικα, τότε αυτός ωριμάζει πολύ πιο γρήγορα. Οπότε οι προγραμματιστές είναι σε θέση να το βελτιώσουν και να το προσαρμόσουν το λειτουργικό σύστημα στις ανάγκες τους. Έτσι το λογισμικό αυτό θα είναι πιο εύκαμπτο και μιας καλύτερης ποιότητας από το λογισμικό που έχει αναπτυχθεί μέσα σε ένα κλειστό κύκλο προγραμματιστών υπεύθυνων για την ανάπτυξη λειτουργικών συστημάτων.

Με την πάροδο του χρόνου τα πλεονεκτήματα ενός open source λειτουργικού συστήματος γίνονται όλο ένα και πιο κατανοητά στον εμπορικό κόσμο και όλο και αυξάνεται η χρήση λειτουργικών συστημάτων, όπως το LINUX, στον εμπορικό κόσμο. Δηλαδή κάτι που είχε γίνει αντιληπτό εδώ και 20 χρόνια στον ακαδημαϊκό κόσμο τώρα αρχίζει να υιοθετείται και από άλλους, πιο συντηρητικούς και δύσκολους χώρους.

Σήμερα το LINUX έχει ξεφύγει από το στάδιο που ήταν καθαρά ένα ακαδημαϊκό σύστημα χρήσιμο μόνο για ένα περιορισμένο αριθμό ανθρώπων που είχαν ένα σημαντικό τεχνικό υπόβαθρο. Έχει προχωρήσει για να καλύψει της ανάγκες της σημερινής εποχής και τις απαιτήσεις που αυτή έχει.

4.4 Τα πλεονεκτήματα του *LINUX*;

Πέρα από το πρώτο πλεονέκτημα που θα αναφέρουμε όλα τα υπόλοιπα πλεονεκτήματα πηγάζουν από τα πλεονεκτήματα του UNIX.

- Πρώτα από όλα τα LINUX παρέχονται δωρεάν

Ίσως ένα από τα πιο σημαντικά πλεονεκτήματα που έχουν τα LINUX σε σχέση με άλλα λειτουργικά συστήματα. Μπορεί κάποιος να τα κατεβάσει από το διαδίκτυο ή ακόμα να το αναπροσαρμόσει στις ανάγκες του συστήματος του, μέσα από μια πληθώρα κώδικα που είναι ελεύθερα διαθέσιμος στο διαδίκτυο.

Η άδεια που χρησιμοποιείται είναι η GNU δημόσια άδεια (GPL). Η άδεια λέει πως ο οποιοσδήποτε έχει δικαίωμα να αλλάξει τον κώδικα και τελικά να ανακατανείμει μια αλλαγμένη έκδοση, υπό τον όρο ότι ο κώδικας είναι ακόμα διαθέσιμος μετά από την ανακατανομή.

- Τα LINUX είναι συμβατά με όλες τις πλατφόρμες Υλικού

Για παράδειγμα ένας προμηθευτής που θέλει να πωλήσει έναν νέο τύπο υπολογιστή και που δεν ξέρει σε πιο είδος λειτουργικού συστήματος (OS) η νέα μηχανή του θα τρέξει μπορεί να πάρει έναν πυρήνα Linux και να τον τροποποιήσει ώστε να είναι κατάλληλος για το νέο τύπο υπολογιστή. Αφού τα LINUX είναι ελεύθερα και ανοικτά

- Τα LINUX είναι σχεδιασμένα να λειτουργούν συνέχεια

Το παραπάνω, ίσως να μην γίνεται απολύτως κατανοητό με την πρώτη ματιά. Αυτό που θέλει να τονίσει είναι πως τα LINUX όπως και τα UNIX άλλωστε, μπορούν να λειτουργούν συνέχεια χωρίς να χρειάζεται επανεκκίνηση (rebooting) όπως σε άλλα λειτουργικά συστήματα. Αυτός είναι και ο λόγος που πολλές εργασίες (tasks), προγραμματίζονται να γίνονται κατά τη διάρκεια της νύχτας ώστε τη μέρα τα μηχανήματα να είναι σε μεγαλύτερη διαθεσιμότητα αφού όπως είναι φυσικό τη μέρα τα μηχανήματα είναι πιο απασχολημένα. Επίσης γι αυτό το λόγο σε μηχανήματα που ο άνθρωπος δεν έχει το χρόνο ή τη δυνατότητα να

βρίσκεται από «πάνω», επιλέγουν να τρέχουν σε LINUX. Αυτό είναι άλλο ένα πλεονέκτημα που πηγάζει από τα βασικά πλεονεκτήματα που έχουν τα UNIX.

➤ Το Linux είναι ασφαλές και ευπροσάρμοστο:

Το πρότυπο ασφάλειας που χρησιμοποιείται για το LINUX είναι βασισμένο στο πρότυπο ασφαλείας που χρησιμοποιείται για το UNIX, το οποίο είναι γνωστό για την ποιότητα που αυτό παρέχει και το πόσο γερό είναι. Αλλά το LINUX δεν είναι μόνο κατάλληλο για τη χρήση ως οχυρό ενάντια στις εχθρικές επιθέσεις από το Διαδίκτυο, αλλά μπορεί να προσαρμοστεί εξίσου σε άλλες καταστάσεις, που χρησιμοποιούν τα ίδια υψηλά πρότυπα για την ασφάλεια.



Χαρακτηριστικό παράδειγμα της αποτελεσματικότητας και της σταθερότητας του LINUX είναι η δημοσίευση της NASA που πρόσφατα ολοκλήρωσε ένα επίσημο project με το όνομα **FlightLinux Project** το οποίο αξιοποιεί τα παραπάνω και όχι μόνο χαρακτηριστικά του LINUX στη διαχείριση και τον έλεγχο επανδρωμένων διαστημικών αεροσκαφών.

4.5 Τα μειονεκτήματα του LINUX;

Παραπάνω, αναφέραμε τα βασικά πλεονεκτήματα που παρέχει το LINUX. Θα αναρωτηθεί κάποιος: Αφού αυτό το λειτουργικό έχει τόσα πολλά και σημαντικά πλεονεκτήματα γιατί δεν έχει κατακλύσει τον κόσμο και γιατί δεν έχει γίνει το βασικό λειτουργικό σύστημα που να το χρησιμοποιεί η πλειοψηφία του κόσμου. Η απάντηση είναι απλή. Όπως όλα τα πράγματα έτσι κι αυτό έχει τα πλεονεκτήματά του, αλλά έχει και κάποια μειονεκτήματα που το εμποδίζουν να γίνει το λειτουργικό σύστημα που θα το χρησιμοποιεί όλος ο κόσμος. Σε αυτό το κεφάλαιο, θα αναφέρουμε σε κάποια από αυτά.

- ✦ Υπάρχουν πάρα πολλές διαφορετικές εκδόσεις.

Όπως οι Ρωμαίοι ήδη έχουν πει: Όσοι περισσότεροι άνθρωποι, τόσες περισσότερες απόψεις και ιδέες. Κάτι αντίστοιχο συμβαίνει και με τα LINUX. Με τη πρώτη ματιά, ο αριθμός των εκδόσεων που υπάρχουν για τα LINUX, μπορεί να τρομοκρατήσουν κάποιον ή να τον κάνουν να γελάσει, ανάλογα από την οπτική γωνία που θα το δει. Επίσης όμως σημαίνει πως ο κάθε ένας μπορεί να βρει ακριβώς αυτό που ζητάει και δεν χρειάζεται να είσαι ο expert για να δεις πιο σου ταιριάζει.

Αν ρωτήσεις κάποιον χρήστη LINUX ποια είναι η καλύτερη έκδοση, θα σου πει πως η καλύτερη είναι αυτή που χρησιμοποιεί αυτός. Τώρα δημιουργείται το ερώτημα: ποια είναι η καλύτερη έκδοση και ποια να διαλέξεις. Εδώ έρχονται τα πράγματα να σε ηρεμήσουν. Όλες οι εκδόσεις πάνω κάτω, περιέχουν το ίδιο σύνολο από τα βασικά πακέτα. Στα πιο πάνω επίπεδα γίνεται η διαφοροποίηση. Για παράδειγμα, τα TurboLinux είναι πιο κατάλληλα για τις μικρομεσαίες επιχειρήσεις, τα RedHat για servers και τα Suse για τερματικούς σταθμούς εργασίας. Η καλύτερη στρατηγική για την επιλογή της κατάλληλης έκδοσης, είναι να δοκιμάσει κάποιος μερικές εκδόσεις του LINUX και να αποφασίσει ο ίδιος ποια είναι η κατάλληλη γι αυτόν. Δυστυχώς, δεν έχουν όλοι το χρόνο για να προβούν σε μια τέτοια διαδικασία. Οι πιο γνωστές εκδόσεις είναι:

- Mandrake
- Red Hat (Fedora)
- Debian
- Suse

- ✦ Τα LINUX, δεν είναι τόσο φιλικά προς τον χρήστη και είναι δύσχρηστα για τους αρχάριους χρήστες.

Τα LINUX, δεν είναι τόσο φιλικά προς το χρήστη σε σχέση με άλλα λειτουργικά συστήματα. Έτσι στους νέους χρήστες υπολογιστών τους φαίνεται αρκετά δύσκολο η χρήση ενός υπολογιστή. Αυτό οφείλεται στο ότι τα LINUX δεν δώσανε τόσο έμφαση στην ευκολία χρήσης όσο κυρίως στην δυνατότητα των χρηστών να επεμβαίνουν σε κομμάτια του λειτουργικού που σε άλλα λειτουργικά συστήματα αυτό είναι αδύνατο. Βέβαια, τα τελευταία χρόνια, έχει σημειωθεί σημαντική βελτίωση στην

ευκολία χρήσης. Και μάλιστα οι τελευταίες εκδόσεις υποστηρίζουν παραθυρικό περιβάλλον. Μπορούμε τέλος να πούμε πως από την πλευρά της ευκολία χρήσης, τα LINUX βρίσκονται ανάμεσα στα UNIX και WINDOWS. Δηλαδή είναι σαφώς πιο εύκολα στη χρήση από ότι τα UNIX, αλλά δεν είναι τόσο εύκολα στη χρήση όσο τα WINDOWS.

Αυτό είναι ένα από τα πιο σημαντικά μειονεκτήματα, αφού πολλοί χρήστες βλέπουν με φόβο την επιλογή που λέγεται: LINUX και πολλοί επιχειρηματίες σκέφτονται το τυχόν κόστος εκμάθησης των υπαλλήλων τους. Έτσι τα LINUX, παρόλο που σου «δίνουν» ότι τα WINDOWS και ακόμα περισσότερα, δεν έχουν γίνει η κύρια επιλογή των χρηστών ηλεκτρονικών υπολογιστών. Βασικά η πλειοψηφία του κόσμου, ξεκινά με WINDOWS, εξοικειώνεται με τη χρήση H/Y και στο επόμενο στάδιο και με την προϋπόθεση ότι «ζητά» περισσότερα από έναν υπολογιστή, προχωράει στο στάδιο την επιλογής που λέγεται: LINUX.

4.6 Software και εντολές που υποστηρίζουν τα LINUX

4.6.1 Software

➤ Βασικές εντολές του LINUX:

ls, tr, sed, awk και άλλες εντολές που προέρχονται από τα UNIX (μπορεί να διαφέρει η ονομασία).

➤ Εργαλεία ανάπτυξης(Development Tools):

gcc, gdb, make, bison, flex, perl, rcs, cvs, prof

➤ Γλώσσες Προγραμματισμού και περιβάλλοντα:

C, C++, Objective C, Java, Modula-3, Modula-2, Oberon, Ada95, Pascal, Fortran, ML, scheme, Tcl/tk, Perl, Python, Common Lisp, και άλλες

- Πολλαπλό Γραφικό Περιβάλλον:
GNOME and KDE (desktops), X11R6 (XFree86 3.x), X11R5 (XFree86 2.x), MGR.
- Κειμενογράφοι (Editors):
GNU Emacs, XEmacs, MicroEmacs, jove, ez, epoch, elvis (GNU vi), vim, vile, joe, pico, jed, και άλλοι.
- Πυρήνες(Shells):
bash (POSIX sh-compatible), zsh (includes ksh compatibility mode), pdksh, tcsh, csh, rc, es, ash (mostly sh-compatible shell used as /bin/sh by BSD), και άλλοι.
- Τηλεπικοινωνίες(Telecommunication):
PPP, UUCP, SLIP, CSLIP, full TCP/IP επικοινωνιακά εργαλεία: kermi, szrz, minicom, pcomm, xcomm, term, Seyon, και άλλα.
- Mail:
C-news, innd, trn, nn, tin, smail, elm, mh, exmh, pine, mutt, και άλλα.
- Επεξεργαστές Κειμένου(Textprocessing):
TeX, groff, doc, ez, LyX, Lout, Linuxdoc-SGML, και άλλους.
- Παιχνίδια(Games):
Όσο και αν φαίνεται αστείο, τα LINUX, υποστηρίζουν πληθώρα παιχνιδιών τα οποία είναι και πολύ δημοφιλή.

Όλα τα προαναφερθείσα προγράμματα, που πρέπει να πούμε πως δεν είναι ούτε το 1% από αυτά που είναι διαθέσιμα, διατίθενται ελεύθερα και δωρεάν στην αγορά.

4.6.2 Εντολές (Commands)

Στο κεφάλαιο αυτό, θα αναφέρουμε τις πιο σημαντικές εντολές, αυτές κυρίως που θα βοηθήσουν τους αρχάριους και όχι μόνο χρήστες, να ξεκινήσουν να χρησιμοποιούν τα LINUX.

Πριν προχωρήσουμε στις εντολές, θα έπρεπε να διευκρινίσουμε πως τα LINUX έχουν γραφικό περιβάλλον, αλλά έχει και terminals που είναι κάτι σαν το αντίστοιχο command prompt των windows. Οι παρακάτω εντολές αφορούν όπως είναι φανερό τα terminals και όχι το γραφικό περιβάλλον.

Τέλος πρέπει να δοθεί προσοχή στο ότι τα LINUX έχουν ένα λογαριασμό που ονομάζεται **root**. Μοιάζει με τον χρήστη Administrator των Windows. Ο χρήστης που μπαίνει στο μηχάνημα σαν root έχει τον πλήρη έλεγχο του μηχανήματος και μπορεί να επέμβει από αρχεία μέχρι των πυρήνα του λειτουργικού και ό,τι αυτό συνεπάγεται. Για πολλές λειτουργίες, είναι αναγκαία η είσοδος σαν root, παρόλα αυτά, συνιστάται στο να αποφεύγεται η είσοδος σαν root γιατί με μία λάθος κίνηση μπορεί να δημιουργηθούν σοβαρότατα προβλήματα στη λειτουργία του λειτουργικού.

Στη συνέχεια αναφέρουμε τις βασικές εντολές του LINUX.

- ✦ Για να δούμε τα αρχεία που περιέχει ένας φάκελος (directory), χρησιμοποιούμε την εντολή **ls** που είναι ίσως η πιο χρήσιμη και πιο χρησιμοποιούμενη εντολή στα LINUX και προκύπτει από την αγγλική λέξη list.
 - ✦ Για να δουλέψουμε με ένα αρχείο ή φάκελο, είναι απαραίτητο να πούμε που αυτό βρίσκεται και που να «ψάξει» να το βρει. Αυτό γίνεται με τον εξής τρόπο:
 - ./ Όταν πρόκειται για το μονοπάτι που βρισκόμαστε ήδη
 - ../ Όταν πρόκειται για ένα μονοπάτι πιο πίσω από αυτό που βρισκόμαστε
 - / Όταν πρόκειται για το root μονοπάτι που είναι και το πιο ψηλό επίπεδο(top level)
- Όταν δεν αναφέρουμε το μονοπάτι τότε εννοείται πως μιλάμε για αρχεία που βρίσκονται στο ίδιο μονοπάτι με αυτό που βρισκόμαστε.

✦ Αντιγραφή αρχείων

Για την αντιγραφή αρχείων χρησιμοποιούμε την εντολή **cp** που προκύπτει από την αγγλική λέξη copy. Παρακάτω θα δώσουμε κάποια παραδείγματα για το πώς χρησιμοποιείται η εντολή cp.

- `cp file1.xxx file2.xxx`

Αυτή η εντολή δημιουργεί ένα αντίγραφο του file1 με το όνομα file2 στο ίδιο μονοπάτι.

- `cd /home/tmp/file1.xxx cd /home/other_file/file2.xxx`

Η παραπάνω εντολή κάνει ό,τι και η προηγούμενη με τη διαφορά ότι ορίζουμε τα μονοπάτια που βρίσκεται το αρχείο που θέλουμε να αντιγράψουμε και το μονοπάτι στο οποίο θέλουμε να αποθηκεύσουμε το αντίγραφο.

- `cd /home/tmp/file1.xxx cd /home/other_file/`

Η εντολή αυτή, είναι ίδια με την παραπάνω, μόνο που δεν έχουμε δώσει το όνομα που θέλουμε να έχει το αντίγραφο. Τότε τα LINUX δίνουν το ίδιο όνομα (file1.xxx) στο αντίγραφο.

- `cd -R /home/tmp/haris/ /home/askisi/alexis/`

Αντιγράφει το φάκελο haris που βρίσκεται στο μονοπάτι /home/tmp/ στο φάκελο alexis που βρίσκεται στο μονοπάτι /home/askisi/.

✦ Διαγραφή αρχείων

Με τον ίδιο τρόπο που αντιγράφουμε αρχεία, μπορούμε και να τα διαγράψουμε. Σε αυτή την περίπτωση, χρησιμοποιούμε την εντολή **rm** που προκύπτει από την αγγλική λέξη remove. Μετά την πληκτρολόγηση της εντολής και αφού πατηθεί το enter, για λόγους ασφαλείας ρωτάτε ο χρήστης αν όντως επιθυμεί την διαγραφή του συγκεκριμένου αρχείου. Στη συνέχεια δίνουμε κάποια παραδείγματα χρήσης της εντολής rm.

- `rm file1`
- `rm file1.xxx`
- `rm /home/tmp/askisi1.xxx`
- `rm -r askisi` για την διαγραφή ενός φακέλου (Directory).

➤ Δημιουργία Φακέλων

Για να δημιουργήσουμε ένα φάκελο (Directory), χρησιμοποιούμε την εντολή **mkdir** που προκύπτει από τις αγγλικές λέξεις **make** **directory**. Παρακάτω, δίνεται ένα παράδειγμα στο οποίο θέλουμε να δημιουργήσουμε το φάκελο askisi στο μονοπάτι που βρισκόμαστε.

- `mkdir askisi`

➤ Αλλαγή μονοπατιού

Πληκτρολογώντας **pwd** σε μια γραμμή εντολών μας δείχνει σε πιο μονοπάτι βρισκόμαστε. Κάνοντας χρήση της εντολής **cd..** πάμε ένα μονοπάτι πάνω. Για παράδειγμα, αν βρισκόμαστε στο μονοπάτι `/home/tmp/file1/file2` και πληκτρολογήσουμε `cd..`, θα πάμε στο μονοπάτι `/home/tmp/file1/`. Αν πληκτρολογήσουμε `cd`, θα πάμε στο home directory κατευθείαν χωρίς να χρειάζεται να πληκτρολογήσουμε δύο φορές `cd..`. Πληκτρολογώντας `cd /home/tmp/file1`, πάμε στο φάκελο ή στο αρχείο με το όνομα `file1`. Αν το αρχείο μας βρίσκεται στο directory που βρισκόμαστε, τότε δεν είναι ανάγκη να γράψουμε το μονοπάτι που βρίσκεται, αρκεί να γράψουμε μόνο το όνομα του `cd file1`.

➤ Ανοιγμα pdf αρχείων στο LINUX

- `acroread file1`

➤ Vi Editor

Ένας πολύ χρήσιμος κειμενογράφος, με πολλές δυνατότητας είναι ο vi.

- Ξεκίνημα

Πληκτρολογώντας την εντολή `vi file1` σε ένα terminal, ανοίγεις ένα κειμενογράφο με το όνομα `file1` που αποθηκεύεται στο μονοπάτι που βρίσκεσαι.

- Όπως όλοι οι κειμενογράφοι, έτσι και ο `vi` έχει ένα κομμάτι με εντολές και ένα κομμάτι για γράψιμο. Για να μπεις στο κομμάτι με τις εντολές, αρκεί να πατήσεις `esc`. Εάν βρίσκεσαι στο κομμάτι των εντολών και πατήσεις `esc` τότε θα ακουστεί από τον υπολογιστή ένα `beep`. Όταν ανοίγεις έναν `vi` editor, βρίσκεσαι στο κομμάτι εντολών και για να περάσεις στο κομμάτι του γραψίματος πρέπει να πατήσεις **a** ή **i**. Η διαφορά τους είναι πως πατώντας **a** αρχίζεις να γράφεις ακριβώς μετά από εκεί που βρίσκεται ο δρομέας(κέρσορας) ενώ πατώντας **i** αρχίζεις να γράφεις ακριβώς πριν τον δρομέα. Υπάρχει μια σειρά από χαρακτήρες που αν τους πατήσεις όταν βρίσκεσαι σε κομμάτι εντολών περνάς στο κομμάτι γραψίματος με μια ιδιότητα.
- Τελειώνοντας για να βγεις και να σώσεις το περιεχόμενο, πατάς `esc` οπότε περνάς στο κομμάτι εντολών πατάς **w** (**write**) για να σώσεις το περιεχόμενο και εμφανίζεται στο κάτω μέρος της οθόνης και μετά **q**: (**quit**) για έξοδο από τον κειμενογράφο.

✦ Μετακίνηση Αρχείων – Φακέλων

Λόγω της δομής του LINUX και του τρόπου πρόσβασης στα αρχεία, η μετακίνηση αρχείων, ουσιαστικά είναι η ίδια με τη μετονομασία. Χρησιμοποιούμε την εντολή **mv** που προκύπτει από το `move`.

- `mv file1 file2`
- `mv /home/temp/file1 /home/temp/askisi/file2`

✦ Τα LINUX παρέχουν πληροφορίες για τις εντολές.

Πληκτρολογώντας **man** και το όνομα της εντολής, το σύστημα σου δίνει πληροφορίες για το τι κάνει η συγκεκριμένη εντολή και πως αυτή συντάσσεται.

- `man cp`

Για να βγούμε στη συνέχεια από τη βοηθητική σελίδα πρέπει να πατήσουμε `q`.

- ✦ Τα LINUX υποστηρίζουν την απομακρυσμένη πρόσβαση σε υπολογιστή. Υπάρχουν αρκετοί τρόποι για να γίνει αυτό, ο πιο συνηθισμένος είναι με τη χρήση της εντολής **SSH**. Προκύπτει από το Secure Shell και οι πιο πολλοί το χρησιμοποιούνε γιατί είναι ένας από τους ασφαλέστερους τρόπους λόγω της κρυπτογράφησης των στοιχείων. Ο SSH μπορεί να χρησιμοποιήσει διάφορες διαφορετικές μορφές κρυπτογράφησης, από 56 έως 1024 bits.

- `ssh username@classes.cs.uchicago.edu`

- ✦ Για τη μεταφορά αρχείων μεταξύ υπολογιστών, τα LINUX δεν υποστηρίζουν το γνωστό σε όλους ftp. Και ο λόγος είναι πως δεν είναι τόσο ασφαλής μέθοδος. Όπως γίνεται κατανοητό μέχρι τώρα, τα LINUX δίνουν μεγάλη σημασία στο θέμα της ασφάλειας. Αντί λοιπόν για ftp τα LINUX χρησιμοποιούν το scp.

- `scp user@remote-computer:/remote/path/remote-file
/local/path/local-file`

Πρέπει να αναφέρουμε εδώ πως η μεταφορά αρχείων μέσω του scp γίνεται μόνο μέσα από υπολογιστές που τρέχουν UNIX ή LINUX και όχι με υπολογιστές που τρέχουν σε Windows.

5 Bluetooth stacks για LINUX

Για να πραγματοποιήσουμε μια σύνδεση με το διαδίκτυο μέσω του υπολογιστή, χρειαζόμαστε έναν προσαρμοστή δικτύου (network adapter), ένα λογισμικό οδηγών ταιριάσματος και το λογισμικό πρωτοκόλλου (γνωστό και ως TCP/IP stack) προκειμένου να επικοινωνήσουμε με άλλες συσκευές. Με Bluetooth η περίπτωση είναι λίγο-πολύ παρόμοια. Χρειαζόμαστε έναν προσαρμοστή Bluetooth, το λογισμικό οδηγών(drivers) και μια λίστα πρωτοκόλλου Bluetooth. Καθώς οι περισσότερες Bluetooth συσκευές αναγνωρίζονται ως σειριακές, δεν υπάρχει συνήθως κανένας ειδικός οδηγός που να απαιτείται. Έτσι για Linux υπάρχουν διάφορες stacks Bluetooth που μπορούμε να επιλέξουμε. Αναφέρουμε ενδεικτικά τις υπάρχουσες:

5.1 OpenBT

Η OpenBT ήταν η πρώτη διαθέσιμη Bluetooth stack για τα Linux. Αναπτύχθηκε αρχικά από την Axis Communications Inc. και είναι τώρα ένα ανοιχτό πρόγραμμα πηγής που φιλοξενείται στη Sourceforge.

Project Homepage: <http://developer.axis.com/software/bluetooth/>

Latest official version: 0.0.5

Download: <http://sourceforge.net/projects/openbt/>

5.2 BlueZ

Η BlueZ αναπτύχθηκε αρχικά από την Qualcomm και αργότερα έγινε ανοιχτό πρόγραμμα πηγής. Από τότε η έκδοση 2.4.6 και νεότερες συμπεριλαμβάνονται σε όλες τις διανομές των Linux. Αποτελεί μαζί με την Affix την πιο ‘δημοφιλή’ stack για τα Linux

Project Homepage: <http://bluez.sourceforge.net/>

Latest official version: XXX

Download: <http://bluez.sourceforge.net/>

5.3 *BlueDrekar*

Το BlueDrekar είναι ένα πρόγραμμα λογισμικού από την IBM που περιλαμβάνει μία πλήρη σωρό Bluetooth. Μόνον ο οδηγός μεταφορών(transport driver) είναι ανοικτή πηγή.

Project Homepage: <http://www.alphaWorks.ibm.com/tech/bluedrekar>

Latest official version: 1.3.0

Download: <http://www.alphaWorks.ibm.com/tech/bluedrekar>

6 Affix

6.1 Εισαγωγή

Η Affix αναπτύχθηκε από μια ερευνητική ομάδα της Nokia και είναι ένα δυνατό και ολοκληρωμένο πρόγραμμα ανοικτής πηγής. Επιτρέπει τη σύνδεση με οποιαδήποτε συσκευή Bluetooth όπως κινητά τηλέφωνα, pocket pcs, και υπολογιστές. Το affix έχει δοκιμαστεί σε i386, ARM, και Power PC πλατφόρμες.

Project Homepage: <http://affix.sourceforge.net>

Latest official version: 3.1.0

Download: <http://affix.sourceforge.net>

Για τους σκοπούς της διπλωματικής μας εργασίας επιλέχθηκε η χρήση του Affix. Τα κριτήρια για την επιλογή του είναι η συμβατότητα του με όλες τις Nokia συσκευές καθώς και ότι είναι μια ολοκληρωμένη εφαρμογή που καλύπτει όλες τις απαιτήσεις αυτού του είδους της ασύρματης τεχνολογίας. Ακολουθεί μια αναλυτική περιγραφή της Affix στοίβας και των δυνατοτήτων του.

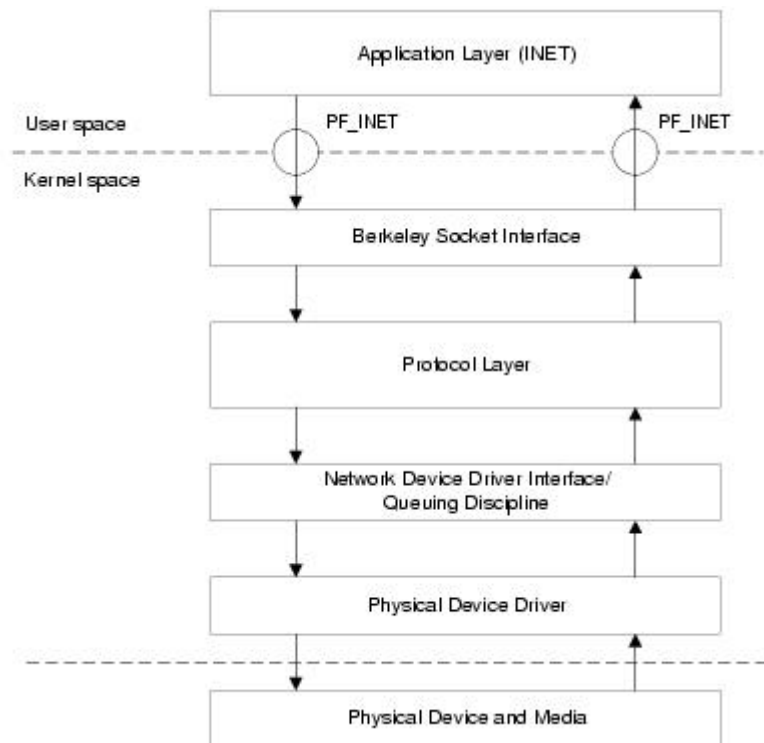
6.2 Εισαγωγή στο Affix

Πρώτα πρέπει να γίνει μια μικρή αναφορά στην σύλληψη της ιδέας αρχιτεκτονικής του πυρήνα, και ακολουθεί η περιγραφή της αρχιτεκτονικής του affix και των βασικότερων συστατικών του.

6.2.1 Αρχιτεκτονική του πυρήνα των Linux

Το λειτουργικό σύστημα Linux εφαρμόζει την τυποποιημένη υποδοχή(socket) API του Μπέρκλεϋ, η οποία έχει την προέλευσή της στις εξελίξεις Unix BSD (4.2/4.3/4.4 BSD) RUSL99.

Η αρχιτεκτονική υποσυστημάτων δικτύων Linux παρουσιάζεται στο παρακάτω σχήμα 1. Τα τυποποιημένα συστατικά του υποσυστήματος δικτύων Linux είναι διεπαφή υποδοχών του Berkley (socket interface) και η διεπαφή οδηγών συσκευών δικτύων(Network device driver interface).



ΠΙΝΑΚΑΣ 6.2.1

Η διεπαφή υποδοχών του Berkley επιτρέπει σε όλα τα προγράμματα διαθέσιμα στους χρήστες να ανοίξουν ένα άκρο επικοινωνίας με απομακρυσμένες συσκευές. Η υποδοχή (socket) είναι μια αφηρημένη έννοια στο άκρο του διαύλου επικοινωνίας. Η υποδοχή συσχετίζεται άμεσα με το πρωτόκολλο. Συνήθως, το PF_INET χρησιμοποιείται για να συσχετίσει μια υποδοχή με το TCP/IP πρωτόκολλο.

Η Network Device Driver διεπαφή δίνει τη δυνατότητα σε πολλαπλές συσκευές δικτύου να χρησιμοποιούνται την ίδια χρονική στιγμή. Ο οδηγός της συσκευής καταχωρεί την συσκευή στο σύστημα, και κάθε συσκευή έχει έναν κατάλληλο τύπο για να διακρίνει την κατηγορία των συσκευών όπως το Ethernet, PPP, X.25, κτλ. Η network driver διεπαφή περιέχει ένα χρόνο-προγραμματιστή πακέτων μεταξύ των διεπαφών. Ο «προγραμματιστής» αυτός εξασφαλίζει την πειθαρχία στην ουρά αναμονής για διαφορετικούς τύπους συσκευών.

Κάθε πρωτόκολλο καταγράφει τον εαυτό του στην υποδοχή της διεπαφής με ένα κατάλληλο γενικό (οικογενειακό) πρωτόκολλο και στην network device

driver διεπαφή με τον κατάλληλο τύπο πρωτοκόλλου. Κάθε πακέτο που γίνεται δεκτό αποστέλλεται στο κατάλληλο στρώμα πρωτοκόλλου.

Το στρώμα δικτύου είναι ένα αρκετά object-προσανατολισμένο στο σχεδιασμό του, όπως είναι εξάλλου και ο πυρήνας των Linux. Η κύρια δομή του κώδικα στο δίκτυο μας πάει πίσω στις αρχικές υλοποιήσεις του δικτύου και των διεπαφών του Ross Biro and Orest Zborowski. Τα αντικείμενα κλειδιά είναι:

Device or Interface: Μια διεπαφή δικτύου είναι ένας κώδικας προγράμματος για αποστολή και αποδοχή πακέτων δεδομένων. Συνήθως μία διεπαφή χρησιμοποιείται για μια συσκευή όπως είναι μία κάρτα Ethernet, αλλά υπάρχουν συσκευές είναι μόνο λογισμικό όπως π.χ μια loopback συσκευή που χρησιμοποιείται για να στέλνεις δεδομένα στον εαυτό σου.

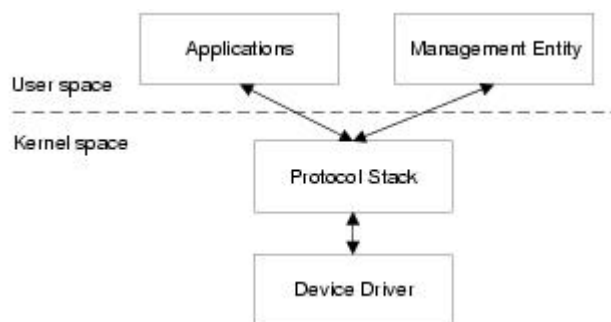
- ✦ Protocol: Κάθε πρωτόκολλο έχει μια αποτελεσματικά διαφορετική γλώσσα επικοινωνίας δικτύου. Μερικά πρωτόκολλα υπάρχουν απλά επειδή οι vendors επιλέγουν να χρησιμοποιούν κατάλληλα περιγράμματα δικτύου, ενώ άλλοι είναι ειδικά σχεδιασμένοι για διαφορετικούς σκοπούς. Μέσα στον πυρήνα των Linux κάθε πρωτόκολλο είναι μια διαφορετική μονάδα (module) κώδικα που παρέχει υπηρεσίες στο στρώμα διεπαφής.
- ✦ Socket: Η υποδοχή είναι μια σύνδεση endpoint που παρέχει στο αρχείο των Unix I/O και υπάρχει σαν file descriptor στο πρόγραμμα του χρήστη. Στον πυρήνα κάθε υποδοχή είναι ένα ζευγάρι δομών το οποίο αντιπροσωπεύει το υψηλό επίπεδο της διεπαφής της υποδοχής και το χαμηλό επίπεδο το πρωτόκολλο διεπαφής.
- ✦ sk_buff: Όλες οι μονάδες προσωρινής αποθήκευσης μνήμης(buffer) που χρησιμοποιούνται από το δίκτυο είναι sk_buffs. Ο έλεγχος για αυτούς τους buffers παρέχεται από χαμηλού επιπέδου διεργασίες(routines) βιβλιοθήκης, οι οποίες είναι διαθέσιμες σε όλο το σύστημα του δικτύου. Τα Sk_buffs παρέχουν όλες τις δραστηριότητες του γενικό buffering και του ελέγχου ροής που χρειάζονται από τα πρωτόκολλα δικτύου.

6.2.2 Affix αρχιτεκτονική

Γενικότερα το Affix software μπορεί να διαιρεθεί σε δύο συνθετικά στοιχεία:

- ✦ Το στοιχείο πρωτοκόλλου(Protocol component) .Αυτό το συστατικό υλοποιεί την στοίβα πρωτοκόλλου.
- ✦ Το στοιχείο διαχείρισης(Management component). Αυτό το συστατικό ελέγχει το στοιχείο πρωτοκόλλου και αποτελεί την ευφυΐα του software.

Η γενική αρχιτεκτονική του λογισμικού περιγράφεται στο παρακάτω σχήμα.



Εικόνα 6.1.2. Affix architectural model

Η φιλοσοφία των Unix (and Linux) σαν λειτουργικό σύστημα είναι ότι αφού ο πυρήνας παρέχει βασικούς μηχανισμούς και οι εφαρμογές χρηστών θα πρέπει να υλοποιούν λογική. Το πρόβλημα αυτής της προσέγγισης είναι σίγουρα η απόδοση του. Σε αυτή την περίπτωση είναι πιθανόν να δημιουργήσουμε κάποια λειτουργικότητα από τον χώρο του χρήστη στον χώρο του πυρήνα, και ο μόνος λόγος για να το κάνουμε αυτό είναι να αυξήσουμε την απόδοση του συστήματος.

Αλλα πλεονεκτήματα-κλειδιά από το να έχουμε περισσότερη λειτουργικότητα στο χώρο του χρήστη είναι η εύκολη υλοποίηση διαδικασιών για τους χρήστες και η καλύτερη σταθερότητα του συστήματος, το οποίο είναι πιο πιθανό να καταρρεύσει από bugs που θα παρουσιαστούν στις εφαρμογές. Στον πυρήνα είναι και πιο πιθανό να έχουμε μια μεγαλύτερη εμβέλεια από debugging δραστηριότητες.

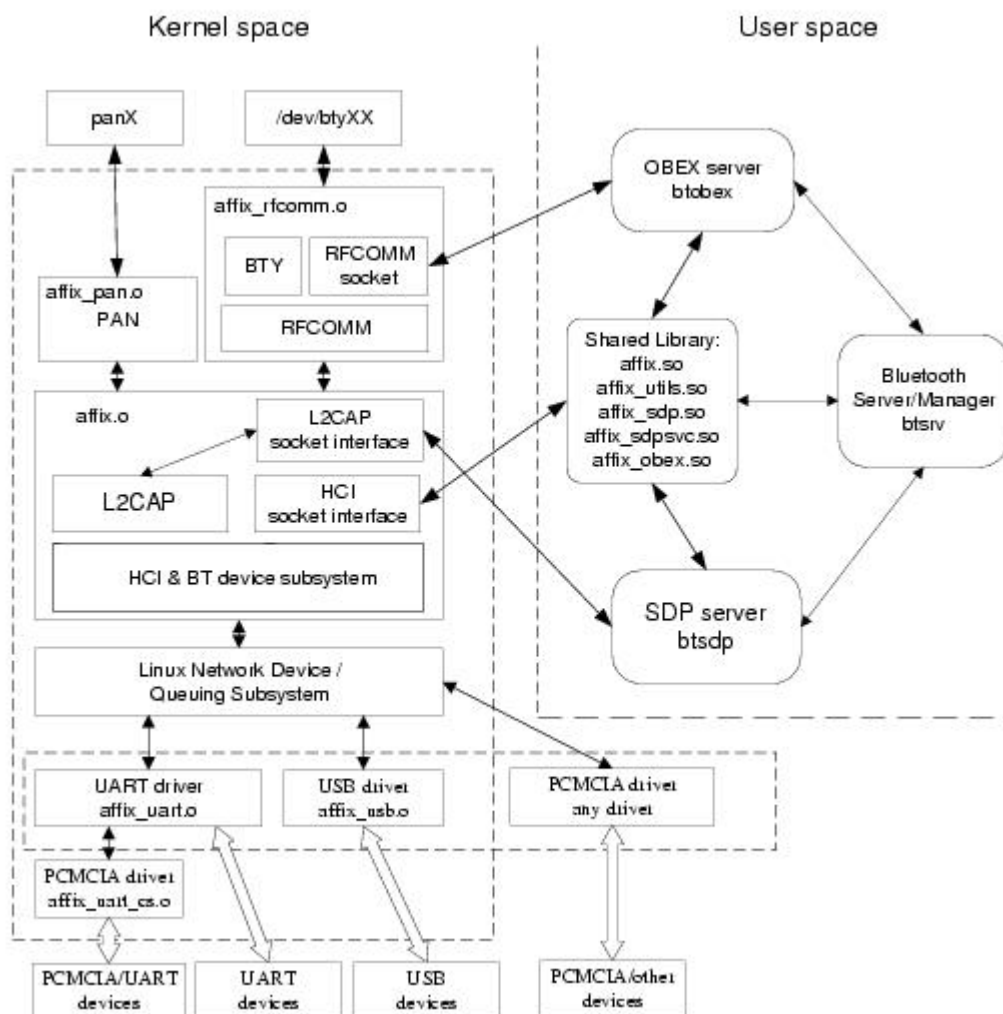
Με βάση λοιπόν την φιλοσοφία του Unix σαν λειτουργικό σύστημα ο πιο κατάλληλος τρόπος για να υλοποιήσεις μια βασική Bluetooth στοίβα πρωτοκόλλου είναι να τρέχει στο χώρο του πυρήνα και το ME να τρέχει στο χώρο των χρηστών. Το Affix ακολουθεί αυτό το σκεπτικό.

Η Bluetooth στοίβα πρωτοκόλλου έχει προκαθορίσει τη λειτουργικότητα του και ορίζεται στις προδιαγραφές του Bluetooth συστήματος. Η στοίβα πρωτοκόλλου από διαφορετικά συστήματα θα πρέπει να έχουν την ίδια λειτουργικότητα, διαφορετικά δεν μπορεί να εγγυηθεί η διαλειτουργικότητα.

Όλη η κίνηση του δικτύου έρχεται στο Affix από το χώρο του πυρήνα. Έτσι η ευνοϊκότερη λύση για απόδοση είναι να υλοποιήσουμε πρωτόκολλα πηγής στο χώρο του πυρήνα.

Το ME έχει να συμπληρώσει την λειτουργικότητα της Bluetooth στοίβας πρωτοκόλλου και μπορεί να έχει μια επιπρόσθετη λειτουργία που θα εξαρτάται από τους σκοπούς του συστήματος. Υλοποιείται σαν μία εφαρμογή στο χώρο των χρηστών. Η απόδοση δεν είναι κάτι κρίσιμο σε αυτό το σημείο. Σε αυτή την περίπτωση είναι πιθανό να ξανά-υλοποιήσουμε το ME χωρίς να χρειάζεται να τροποποιήσουμε τον κώδικα του πυρήνα.
without modifying the kernel space code.

Παρακάτω απεικονίζεται ένα διάγραμμα με τα στοιχεία που απαρτίζουν το affix.



Εικόνα 6.1.2β 2-3. Affix components

6.2.3 Bluetooth and Affix μοντέλα ασφαλείας

Το Bluetooth έχει εσωτερικούς μηχανισμούς ασφαλείας: πιστοποίηση και απόκρυψη. Υπάρχει επίσης έγγραφο αναφορά που περιέχει υπόδειξη και απαιτήσεις πως πρέπει να επιτευχθεί η ασφάλεια. Ένας πρώτος μηχανισμός ασφαλείας βασίζεται σε ένα PIN κώδικα που ο χρήστης θα πρέπει να πληκτρολογήσει για να συνδεθεί με μια συσκευή ή να επιτρέψει άλλες συσκευές να συνδεθούν μαζί του.

Το Affix έχει την απαραίτητη υποστήριξη σε Bluetooth ασφάλεια εισάγοντας υψηλά επίπεδα και κόμβους ασφαλείας τα οποία είναι τα παρακάτω:

- ✦ OPEN. Αυτή είναι μια κατάσταση που επιτρέπει οποιαδήποτε συσκευή να συνδεθεί σε όποια τοπική υπηρεσία επιθυμεί χωρίς πιστοποίηση. (χωρίς PIN).
- ✦ LINK. Αυτή η κατάσταση εκτελεί πράξεις ασφαλείας στο link επίπεδο. Πριν γίνει η σύνδεση στο link επίπεδο (ACL) η συσκευή πρέπει να περάσει έναν έλεγχο(εξαρτάται πάντα από το επίπεδο ασφαλείας).Αφού έχει γίνει η σύνδεση η συσκευή έχει πρόσβαση σε όλες τις υπηρεσίες. Με άλλα λόγια δεν μπορεί να γίνει μια σύνδεση χωρίς να γίνουν οι απαραίτητοι έλεγχοι ασφαλείας
- ✦ SERVICE. Αυτή η κατάσταση εκτελεί πράξεις ασφαλείας στο επίπεδο υπηρεσιών .Η συσκευή μπορεί να δημιουργήσει μια σύνδεση στο link επίπεδο χωρίς ελέγχους ασφαλείας αλλά για να συνδεθεί με μια υπηρεσία πρέπει να περάσει έλεγχο. Αυτή η κατάσταση είναι πιο ευέλικτη σε σχέση με το επίπεδο link επειδή επιτρέπει να καθορίσεις τι επίπεδο ασφάλεια υπηρεσιών μπορεί να περιέχεται στα σημεία ελέγχου.

Το Affix έχει τα παρακάτω επίπεδα ασφαλείας:has following security levels:

- ✦ OPEN. Δεν συμπεριλαμβάνονται πράξεις ασφαλείας.
- ✦ AUTH. Ένας PIN κώδικας απαιτείται για να περάσεις το σημείο ελέγχου.
- ✦ ENCRYPT. Μια σύνδεση απόκρυψης απαιτείται. Αυτό το επίπεδο απαιτεί AUTH επίπεδο.
- ✦ AUTHOR. Μια εξουσιοδότηση απαιτείται για να περάσεις το σημείο ελέγχου. Ο χρήστης πρέπει χειροκίνητα να επιβεβαιώσει την πράξη.(π.χ. να αποδεχτεί την σύνδεση).

Όλοι αυτοί οι μηχανισμοί κάνουν το affix πολύ ασφαλές και ευέλικτο. Για να επιλέξει το επιθυμητό επίπεδο ασφαλείας ο χρήστης πρέπει να θέσει την κατάσταση ασφαλείας και το επίπεδο ασφαλείας για αυτή την κατάσταση. Για παράδειγμα, διαλέγοντας SERVICE κατάσταση πρέπει να απαιτηθεί AUTH πιστοποίηση και να συμπεριληφθεί AUTHOR εξουσιοδότηση.

6.3 Εφαρμογές Προγραμματισμού διεπαφών

Ο σχεδιασμός του Affix επιτρέπει στους χρήστες να υλοποιήσουν εφαρμογές στο επίπεδο του πυρήνα (σαν υπομονάδες του πυρήνα-modules) και στο επίπεδο χρηστών (σαν εφαρμογές). Για αυτό το λόγο το affix εξάγει τα ακόλουθα interfaces:

- ✦ *Kernel programming interface.* Ένα σετ από συναρτήσεις, το οποίο περιέχει HCI, L2CAP, RFCOMM and UART υπομονάδες.
- ✦ *Socket interface.* Ένα πρώτου επιπέδου API που χρησιμοποιείται για να αξιοποιήσει όλα τα πηγαία πρωτόκολλα Bluetooth όπως τα L2CAP και RFCOMM. Χρησιμοποιείται με τον ίδιο τρόπο όπως άλλου τύπου υποδοχές. (π.χ. PF_INET).
- ✦ *Shared libraries.* Ένα σετ από διάφορες συναρτήσεις.

6.3.1 Affix API συστατικά

Τα Affix API συστατικά περιέχουν τις επόμενες οντότητες: ενσωματωμένα στα αρχεία header και στις βιβλιοθήκες. Το προεπιλεγμένο μονοπάτι για εγκατάσταση είναι το /usr/include/affix and /usr/lib αντίστοιχα.

- ✦ header files (/usr/include/affix);
- ✦ libraries (/usr/lib);
- ✦ kernel modules (/lib/modules/`uname -r`/kernel/{net/bluetooth,drivers/bluetooth}).

Πίνακας1. Affix header αρχεία

| Header αρχείο | Περιγραφή |
|--------------------------|---|
| bluetooth.h | Βασικόheader αρχείο. Πρέπει να συμπεριληφθεί σε όλα τα προγράμματα. Περιέχει τους βασικούς ορισμούς του Affix και δηλώσεις συναρτήσεων. Παρέχει διεπαφή μεταξύ τα μέρη του πυρήνα και του χρήστη. |
| hci_types.h | Περιέχει όλους τους ορισμούς τύπου HCI . (ορίζεται σύμφωνα με τις προδιαγραφές του Bluetooth). |
| hci_cmds.h | Περιέχει εντολές υλοποίησης HCI. (ορίζεται σύμφωνα με τις προδιαγραφές του Bluetooth). |
| btcore.h | Περιέχει επιπέδου-χρήστη Affix συγκεκριμένους ορισμούς και δηλώσεις συστήματος. |
| utils.h | Περιέχει δηλώσεις από συνήθη εργαλεία του Affix. |
| sdp.h | Περιέχει ορισμούς τύπου SDP. |
| sdpclt.h | Περιέχει SDP client δηλώσεις διεπαφής. |
| sdpshr.h | Περιέχει SDP server δηλώσεις διεπαφής. |
| obex.h | Περιέχει OBEX client Δηλώσεις διεπαφής. |

Πίνακας 2. Affix Βιβλιοθήκες

| Βιβλιοθήκη | Περιγραφή Description |
|-------------------|---|
| libaffix.so | Η κύρια βιβλιοθήκη του affix.Main Affix library. Περιέχει Affix και Bluetooth υλοποιήσεις πηγής. Όλα τα προγράμματα πρέπει να συνδεθούν με αυτή . |
| libaffix_utils.so | Περιέχει υλοποίηση από γενικού σκοπού συναρτήσεις που χρησιμοποιούνται στο Affix. |
| libaffix_sdp.so | Περιέχει SDP client API υλοποίηση. Κάθε πρόγραμμα που χρησιμοποιεί SDP πρέπει να έχει σύνδεση με αυτή |

| Βιβλιοθήκη | Περιγραφή Description |
|--------------------|---|
| libaffix_sdpsvc.so | Περιέχει SDP server API υλοποίηση. Μόνο προγράμματα με υπηρεσίες καταγραφής συνδέονται με αυτό. |
| libaffix_obex.so | Περιέχει Bluetooth OBEX πλαισίου εργασίας υλοποίηση. Κάθε πρόγραμμα που χρησιμοποιεί OBEX πρέπει να συνδεθεί μαζί της.. |

Πίνακας 3. Affix υπομονάδες πυρήνα

| Υπομονάδα | Περιγραφή |
|-----------------|---|
| affix.o | Affix υπομονάδα πηγή (HCI, L2CAP, και οι διεπαφές των υποδοχών) |
| affix_rfcomm.o | RFCOMM, BTY και οι διεπαφές των υποδοχών |
| affix_pan.o | PAN (BNEP, διεπαφή δικτύου) |
| affix_usb.o | Bluetooth USB οδηγός |
| affix_uart.o | Bluetooth UART (H4, BCSP, TLP) οδηγός |
| affix_uart_cs.o | PCMCIA οδηγός για UART κάρτες |

6.3.2 Υποδοχή διεπαφής (Socket interface)

Η υποδοχή διεπαφής παρέχει μια βασική καλά οργανωμένη προσέγγιση για να προσπελάσεις τους πόρους του δικτύου του πυρήνα. Το affix εισάγει στο σύστημα μια νέα οικογένεια πρωτοκόλλων την PF_AFFIX και νέα οικογένεια με διευθύνσεις υποδοχών "struct sockaddr_affix":

<affix/bluetooth.h> πρέπει να συμπεριληφθεί στο source αρχείο.

Τυπικό "socket" κάλεσμα συστήματος χρησιμοποιείται για να δημιουργήσει ένα socket από την οικογένεια PF_AFFIX :

```
int socket(PF_AFFIX, int type, int protocol);
```

όπου η *type* παράμετρος εξαρτάται από την *protocol* παράμετρο που παίρνει τις ακόλουθες τιμές:

- ✦ BTPROTO_HCIACL. Δημιουργεί HCI ACL socket. Έγκυρος τύπος: SOCK_SEQPACKET.
- ✦ BTPROTO_HCISCO. Δημιουργεί HCI SCO socket. Έγκυρος τύπος : SOCK_SEQPACKET.
- ✦ BTPROTO_L2CAP. Δημιουργεί L2CAP socket. Έγκυρος τύπος : SOCK_STREAM, SOCK_SEQPACKET.
- ✦ BTPROTO_RFCOMM. Δημιουργεί RFCOMM socket. Έγκυρος τύπος : SOCK_STREAM.

6.3.2.1 Δομές δεδομένων

Η διεπαφή της υποδοχής έχει μία μόνο δομή δεδομένων όπως κάθε οικογένεια πρωτοκόλλων στο Unix-συγγένεια με `sockaddr`. Η δομή έχει το όνομα `"sockaddr_affix"` και έχει το ακόλουθο πρωτότυπο:

```
struct sockaddr_affix {  
    sa_family_t    family;  
    BD_ADDR        bda;  
    uint16_t       port;  
    BD_ADDR        local;  
};
```

Όλα τα παραπάνω πεδία εκτός από το `port` (θύρα) έχουν το ίδιο νόημα για όλα τα πρωτόκολλα στη οικογένεια του `PF_AFFIX`. Το `"port"` πεδίο έχει διαφορετική σημασία για διαφορετικά πρωτόκολλα.:

- ✦ BTPROTO_L2CAP: `"port"` ορίζει PSM τιμή.
- ✦ BTPROTO_RFCOMM: `"port"` ορίζει το νούμερο του καναλιού του server

- ✦ BTPROTO_HCIACL και BTPROTO_HCISCO: "port" πεδίο δεν χρησιμοποιείται.

6.3.2.2 Ομάδα συναρτήσεων

Η οικογένεια υποδοχών PF_AFFIX επιτρέπει να χρησιμοποιήσεις τις τυπικές συναρτήσεις όπως είναι η connect(), bind(), listen(), accept(), send(), recv() και έχει επιπρόσθετη Bluetooth και Affix συγκεκριμένη επέκταση. Περισσότερες λεπτομέρειες για κάλεσμα υποδοχών μπορεί να βρεθεί στις manual σελίδες του Unix.

Η PF_AFFIX επέκταση πρακτικά υλοποιείται ως ioctl() και set(get)sockopt() system καλέσματα, αλλά για πιο άνεση περιβάλλεται στις καλώς ονομαζόμενες συναρτήσεις.

Πίνακας 4. PF_AFFIX extra API

| Function | Applicability | Purpose |
|------------------------------------|-----------------------------------|---|
| int l2cap_setmtu(int fd, int mtu); | BTPROTO_L2CAP | Θέτει ένα μέγιστο μέγεθος πακέτου που μπορεί να επιτραπεί σε αυτή την υποδοχή. Αυτή η συνάρτηση χρησιμοποιείται πριν καλέσεις την connect() ή bind(). |
| int l2cap_getmtu(int fd); | BTPROTO_L2CAP | Δέχεται ένα μέγιστο μέγεθος πακέτου το οποίο μπορεί να αποσταλεί από αυτή την υποδοχή. |
| int hci_getmtu(int fd); | BTPROTO_HCIACL, BTPROTO_HCISCO | Δέχεται ένα μέγιστο μέγεθος πακέτου το οποίο μπορεί να αποσταλεί από αυτή την υποδοχή. |
| int l2cap_ping(int fd, char | BTPROTO_L2CAP | Στέλνει ένα *ping* πακέτο |

| Function | Applicability | Purpose |
|-------------------|---------------|-----------------------------|
| *data, int size); | | και περιμένει για απάντηση. |

Η RFCOMM υποδοχή έχει ένα επιπλέον χαρακτηριστικό: μπορεί να αναπηδήσει σε TTY γραμμή. Το Affix χρησιμοποιεί "/dev/btyXX" ονόματα για αυτό, όπου XX είναι μια καταγεγραμμένη γραμμή. Ειδικές συναρτήσεις που χρησιμοποιούνται σε αυτή :

Πίνακας 5. RFCOMM TTY API

| Συνάρτηση | Σκοπός |
|--|---|
| int rfcomm_open_tty(int fd, int line); | Προσάπτει μια RFCOMM σύνδεση σε μια TTY γραμμή *line*. Εάν η *line* είναι -1 τότε προσδιορίζει μια μη χρησιμοποιημένη tty γραμμή και επιστρέφει το νούμερο της. |
| int rfcomm_close_tty(int line); | Αποσυνδέει μια RFCOMM σύνδεση από μία TTY γραμμή *line* και την κλίνει. |
| Int rfcomm_set_type(int fd, int type); | Αλλάζει τον τύπο της υποδοχής RFCOMM σε: RFCOMM_SOCKET (default) η RFCOMM_BTY. Αυτή η συνάρτηση πρέπει να καλείται αμέσως μετά από την *socket()*.. |

6.3.2.3 Δείγμα κώδικα

Πρακτικά παραδείγματα μπορούν να βρεθούν στον sourceκώδικα του Affix. Παρακάτω ακολουθεί ένα παράδειγμα για να μπορέσει ο αναγνώστης να καταλάβει πως χρησιμοποιούνται οι PF_AFFIX υποδοχές στην πλευρά του client και του server .

Το επόμενο παράδειγμα διευκρινίζει τη χρήση της Affix διεπαφής υποδοχής στην πλευρά του client.

```

#include <stdio.h>

/* Affix includes */
#include <affix/bluetooth.h>

int main(int argc, char *argv[])
{
    struct sockaddr_affix sa;
    int fd, err;

    fd = socket(PF_AFFIX, SOCK_SEQPACKET, BTPROTO_L2CAP);
    if (fd < 0) {
        perror("socket() failed");
        return 1;
    }

    sa.family = PF_AFFIX;
    str2bda(&sa.bda, "00:11:22:33:44:55"); /* connect to that device */
    /* or sa.bda = other_bda; */
    sa.port = 1; /* connect to that port (PSM) */
    sa.local = BDADDR_ANY;
    /*
    to connect through certain device use:
    sa.local = <local device bda>
    */

    err = connect(fd, (struct sockaddr*)&sa, sizeof(sa));
    if (err) {
        perror("connect() failed");
        return 2;
    }

    /*
    here is socket is ready for communication
    any of the standard connection oriented
    transmission/receiving system calls can be used.
    - send(), sendmsg(), write()
    - recv(), recvmsg(), read()
    */
}

```

```

close(fd);

return 0;
}

```

Το επόμενο παράδειγμα διευκρινίζει τη χρήση της Affix διεπαφής υποδοχής στην πλευρά του client.

```

#include <stdio.h>

/* Affix includes */
#include <affix/bluetooth.h>

int main(int argc, char *argv[])
{
    struct sockaddr_affix sa, csa;
    int fd, cfd, err;
    socklen_t csa_len;

    /* create server socket */
    fd = socket(PF_AFFIX, SOCK_SEQPACKET, BTPROTO_L2CAP);
    if (fd < 0) {
        perror("socket() failed");
        return 1;
    }

    sa.family = PF_AFFIX;
    sa.bda = BDADDR_ANY;
    sa.port = 1; /* accept connection to that port (PSM) */
    /*
    to connect through certain device use:
    sa.local = <local device bda>
    */

    /* bind socket to address specified by "sa" parameter */
    err = bind(fd, (struct sockaddr*)&sa, sizeof(sa));
    if (err) {
        perror("bind() failed");
        return 2;
    }
}

```

```

}

/* start listen for connection - kernel will accept connection requests */
err = listen(fd, 5)
if (err) {
    perror("listen() failed");
    return 3;
}

/* accept new connection and get its connection descriptor "cfd" */
csalen = sizeof(csa);
cfd = accept(fd, (struct sockaddr*)&csa, &csa_len);
if (cfd < 0) {
    perror("accept() failed");
    return 4;
}

/*
    here is socket "cfd" is ready for communication
    any of the standard connection oriented
    transmission/receiving system calls can be used.
    - send(), sendmsg(), write()
    - recv(), recvmsg(), read()
*/

close(fd); /* close server socket */
close(cfd); /* close client socket */

return 0;
}

```

6.3.2.4 Επίπεδο ασφαλείας στις υποδοχές

Όλες οι υπηρεσίες λειτουργούν πάνω από τις υποδοχές, ή τουλάχιστον η διαδικασία πραγματοποίησης μιας σύνδεσης γίνεται πάνω από αυτή. Η υπηρεσία ασφαλείας επιτρέπει στην υπηρεσία να θέσει το δικό της επίπεδο ασφαλείας. Αυτό γίνεται μέσα από ένα ειδικό API ασφαλείας.

Πίνακας 6.2.2.43 Socket security API

| Function | Purpose |
|---|--|
| <code>int setsockopt(int fd, SOL_AFFIX, BTSO_SECURITY, int *level, sizeof(int));</code> | Θέτει το επίπεδο ασφαλείας: HCI_SECURITY_OPEN, HCI_SECURITY_AUTH, HCI_SECURITY_ENCRYPT, HCI_SECURITY_AUTHOR. |

6.3.3 HCI API

Το HCI API επιτρέπει στον χρήστη να εκτελεί έναν έλεγχο χαμηλού επιπέδου σε συσκευές Bluetooth ,και να το θέτει σε διαφορετικές καταστάσεις λειτουργίας ,να ανακαλύπτει άλλες Bluetooth συσκευές στην ακτίνα λήψης. Δεν υπάρχουν περιπτώσεις που το HCI παραμένει μη χρησιμοποιημένο. Η πιο συνήθης εργασία είναι να ανακαλύπτει τις συσκευές που βρίσκονται στο πεδίο ακτινοβολίας του.

Η κύρια ιδέα στο HCI API είναι ένας HCI descriptor, ο οποίος στην πραγματικότητα είναι ένα descriptor αρχείο του Unix. Ο HCI descriptor χρησιμοποιείται με μια συνάρτηση καλέσματος.

6.3.3.1 Ομάδα συναρτήσεων

Η HCI API ομάδα συναρτήσεων χωρίζεται σε δύο υπό-ομάδες: ομάδα συναρτήσεων συστήματος (Affix specific) και (Bluetooth device specific) ομάδα συναρτήσεων. Όλα τα ονόματα των συναρτήσεων του Affix έχουν ένα low-case στυλ και όλα τα ονόματα συναρτήσεων του HC έχουν mixed-case στυλ και αρχίζουν από HCI_ ή __HCI_. Οι HC συναρτήσεις υλοποιούνται σύμφωνα με τις προδιαγραφές του συστήματος Bluetooth.

Οι συναρτήσεις του συστήματος του Affix παρουσιάζεται στον παρακάτω πίνακα.

Πίνακας 6.2.3.1 HCI API ομάδα συναρτήσεων

| Πρωτότυπες συναρτήσεις | Περιγραφή |
|---|--|
| <code>int hci_open(char *name);</code> | Ανοίγει την συσκευή "κατά όνομα" και επιστρέφει έναν HCI descriptor. |
| <code>int hci_open_id(int hd);</code> | Ανοίγει τη συσκευή "κατά id" και επιστρέφει έναν HCI descriptor. |
| <code>int hci_get_devs(int *devs);</code> | Αποθηκεύει HCI descriptors από καταγεγραμμένες συσκευές Bluetooth στο σύστημα στο *devs* και επιστρέφει το νούμερο τους. |
| <code>int hci_open_event(void);</code> | Ανοίγει ένα κανάλι που χρησιμοποιείται για να πάρει HCI γεγονότα από κάθε συσκευή και επιστρέφει έναν HCI descriptor. |
| <code>Int hci_listen_event(int fd, __u64 mask);</code> | Θέτει μια μάσκα γεγονότων για μια συγκεκριμένη συσκευή επιτρέπει την αποδοχή γεγονότων του ίδιου τύπου. Η τιμή της μάσκας που ισοδυναμεί με μηδέν καθιστά αδύνατη την λήψη γεγονότων. |
| <code>int hci_recv_event(int fd, void *event, int *hd);</code> | Δέχεται HCI γεγονότα από HCI συσκευές. Τα γεγονότα αποθηκεύονται σε μια μονάδα προσωρινής αποθήκευσης μνήμης έχοντας ως δείκτη ένα "γεγονός" όρισμα και το id συσκευής από την οποία ήρθε το γεγονός και αποθηκεύεται στην μεταβλητή με δείκτη "hd". |
| <code>int hci_exec_cmd(int fd, void *cmd, __u64 mask, int flags, void *event);</code> | Στέλνει HCI εντολές με δείκτη "cmd" στη συσκευή με δείκτη "fd", περιμένει για κάποιο COMMAND_STATUS ή COMMAND_COMPLETE γεγονός και το αποθηκεύει σε έναν buffer με δείκτη "event". Το "mask" όρισμα ορίζει γεγονότα (mask) που είναι για αποδοχή. |

| Πρωτότυπες συναρτήσεις | Περιγραφή |
|---|---|
| <code>int hci_open_mgr(void);</code> | Ανοίγει το εσωτερικό κανάλι ελέγχου του Affix και επιστρέφει έναν descriptor. |
| <code>int hci_recv_msg(int fd, __u8 *param);</code> | Αποδέχεται εσωτερικό μήνυμα ελέγχου του Affix από το κανάλι με δείκτη "fd" και αποθηκεύει δεδομένα στον buffer με δείκτη "param" όρισμα. |
| <code>int hci_add_pin(BD_ADDR *bda, int Length, __u8 *Code);</code> | Αποθηκεύει έναν pin κώδικα στο φράγμα ελέγχου του Affix. Χρησιμοποιείται στην περίπτωση που θα απαιτηθεί. Χρησιμοποιείται από μη ενεργές εφαρμογές. Αυτός ο pin κώδικας χρησιμοποιείται για να πιστοποιήσει τη συσκευή *bda*. |
| <code>int hci_remove_pin(BD_ADDR *bda);</code> | Διαγράφει το pin κώδικα από το φράγμα ελέγχου του Affix. |
| <code>int hci_add_key(BD_ADDR *bda, __u8 key_type, __u8 *key);</code> | Αποθηκεύει ένα κλειδί σύνδεσμο στο φράγμα ελέγχου του Affix. Χρησιμοποιείται σε περίπτωση που η εφαρμογή χρησιμοποιεί κλειδί σύνδεσμο για χρήση στο μέλλον. Το κλειδί σύνδεσμος χρησιμοποιείται και από για να πιστοποιήσει τη συσκευή *bda*. |
| <code>int hci_remove_key(BD_ADDR *bda);</code> | Διαγράφει το κλειδί σύνδεσμο από το φράγμα ελέγχου του Affix. |
| <code>int hci_set_mode(int fd, int mode);</code> | Θέτει συναρτησιακή κατάσταση στον πηγαίο πυρήνα του affix (mode: AFFIX_MODE_PIN - caller που διαχειρίζεται PIN, AFFIX_MODE_KEY – caller που διαχειρίζεται handles key). |
| <code>int hci_get_conn(int fd, BD_ADDR *bda);</code> | Ανακτά τη διαχείριση της HCI ACL/SCO σύνδεσης στη συσκευή *bda*. |
| <code>hci_get_attr(char *name, struct hci_dev_attr *attr);</code> | Διαβάζει τα χαρακτηριστικά της συσκευής: όνομα, id, flags. |

| Πρωτότυπες συναρτήσεις | Περιγραφή |
|--|--|
| <code>hci_get_attr_id(int hd, struct hci_dev_attr *attr);</code> | Διαβάζει τα χαρακτηριστικά της συσκευής: όνομα, id, flags. |
| <code>hci_get_attr_fd(int fd, struct hci_dev_attr *attr);</code> | Διαβάζει τα χαρακτηριστικά της συσκευής: όνομα, id, flags. |
| <code>hci_get_flags(int fd, int *flags);</code> | Διαβάζει τις flags της συσκευής. |
| <code>hci_set_flags(int fd, int flags);</code> | Θέτει τα flags της συσκευής: HCI_FLAGS_UP |
| <code>int hci_set_secmode(int fd, int mode);</code> | Θέτει κατάσταση ασφαλείας: HCI_SECURITY_OPEN, HCI_SECURITY_LINK, HCI_SECURITY_AUTH, HCI_SECURITY_AUTHOR). Αυτές οι καταστάσεις μπορούν επεξεργάζονται λογικά-or'd. |
| <code>int hci_set_pktttype(int fd, int pkt_type);</code> | Θέτει τύπο πακέτου. |
| <code>int hci_set_role(int fd, int role);</code> | Θέτει ρόλους σύνδεσης. |
| <code>char *hcierror(int err);</code> | Επιστρέφει ένα μήνυμα λάθους HCI . |
| <code>char *bda2str(BD_ADDR *bda);</code> | Μετατρέπει Bluetooth διεύθυνση σε αλφαριθμητικοί σειρά χαρακτήρων (string). |
| <code>int str2bda(BD_ADDR *p, char *str);</code> | Μετατρέπει string σε Bluetooth διεύθυνση. |

Οι κυριότερες HC συναρτήσεις παρουσιάζονται στον πίνακα 8. Όλες οι συναρτήσεις δέχονται τουλάχιστον μια παράμετρο - hci descriptor και επιστρέφουν λάθος στον κώδικα, το οποίο παίρνει τις επόμενες τιμές:

- ✦ Μηδέν (0) εάν δεν προκύψει κανένα λάθος.
- ✦ -1 σε περίπτωση που έχουμε λάθος συστήματος, και το λάθος διορθώνεται καταλλήλως.(όπως γίνεται με κάθε κάλεσμα του συστήματος του Linux)
- ✦ πάνω από μηδέν (>0) σε περίπτωση που είναι HCI λάθος.

Η βασική διάταξη της HC συνάρτησης είναι: `int HCI_Xxxx(int fd, param, param, ...)`;

Οι υπόλοιπες παράμετροι των HC συναρτήσεων εξαρτώνται σε μια πραγματική συνάρτηση και ορίζεται σύμφωνα με τις προδιαγραφές του Bluetooth .Η περιγραφή του μπορεί να διαβαστεί κάλλιστα από εκεί.

Ο επόμενος πίνακας παρουσιάζει μόνο ένα μέρος της ομάδας των HC συναρτήσεων που υλοποιούνται στο `affix`.Μια ολοκληρωμένη ομάδα μπορεί να βρεθεί στην `hci_cmds.h`.

Πίνακας-6.2.3.1 HC ομάδα συναρτήσεων

| Συνάρτηση | Περιγραφή |
|---|---|
| <code>int HCI_Inquiry(int fd, __u8 Inquiry_Length, __u8 Max_Num_Responses, INQUIRY_ITEM *Items, __u8 *Num_Responses)</code> | Εκτελεί INQUIRY κατά τη διάρκεια (1.28 * Inquiry_Length) δευτερολέπτων και αποθηκεύει τις πληροφορίες από τις bluetooth συσκευές που βρήκε σε *Items* πίνακα και αριθμεί τις συσκευές σε *Num_Responses*. |
| <code>int HCI_WriteScanEnable(int fd, __u8 Scan_Enable);</code> | Θέτει SCAN κατάσταση (SCAN_OFF, SCAN_INQUIRY, SCAN_PAGE, SCAN_BOTH). |
| <code>int HCI_WriteClassOfDevice(int fd, __u32 Class_of_Device);</code> | Θέτει την κλάση της συσκευής. |
| <code>int HCI_ChangeLocalName(int fd, char</code> | Θέτει όνομα στη συσκευή. Όλες οι |

| Συνάρτηση | Περιγραφή |
|--|--|
| *Name); | απομακρυσμένες συσκευές θα δουν αυτό το όνομα. |
| int HCI_WriteAuthenticationEnable(int fd, __u8 Authentication_Enable); | Επιτρέπει πιστοποίηση. |
| int HCI_WriteEncryptionMode(int fd, __u8 Encryption_Mode); | Θέτει κατάσταση απόκρυψης για όλες τις ACL συνδέσεις: EM_NO, EM_PP (σημείο -με-σημείο), EM_ALL (pp and broadcast). |
| int HCI_ReadBDAddr(int fd, BD_ADDR *bda); | Διαβάζει διεύθυνση από τη συσκευή. |
| int HCI_WritePageTimeout(int fd, __u16 Page_Timeout); | Θέτει χρονική εκπνοή σελίδας. |
| int HCI_WriteLinkPolicy(int fd, CHANDLE Connection_Handle, __u8 Link_Policy_Settings); | Θέτει πολιτική σύνδεσης. |
| | |

6.3.3.2 Δομές δεδομένων

Οι δομές δεδομένων που συσχετίζονται στην HCI διεπαφή ορίζονται στην hci_types.h header.

- struct sockaddr_affix.
- struct sockaddr_affix {
 - sa_family_t family;
 - BD_ADDR bda;
 - uint16_t port;
 - BD_ADDR local; /* local Bluetooth adapter to connect to */
- };
- INQUIRY_ITEM
- typedef struct {

- BD_ADDR bda;
- __u8 PS_Repetition_Mode;
- __u8 PS_Period_Mode;
- __u8 PS_Mode;
- __u32 Class_of_Device:24;
- __u16 Clock_Offset;
- }__PACK__ INQUIRY_ITEM;
- struct hci_dev_attr.
- /* flags mask */
-
- #define HCI_FLAGS_RUNNING 0x00000001
- #define HCI_FLAGS_UP 0x00000002
-
- #define HCI_FLAGS_ROLE 0x000000F0
- #define HCI_ROLE_ALLOW_SWITCH 0x00000000
- #define HCI_ROLE_DENY_SWITCH 0x00000010
- #define HCI_ROLE_REMAIN_SLAVE 0x00000000
- #define HCI_ROLE_BECOME_MASTER 0x00000020
-
- #define HCI_FLAGS_SECURITY 0x0000FF00
- #define HCI_SECURITY_OPEN 0x00000100
- #define HCI_SECURITY_SERVICE 0x00000200
- #define HCI_SECURITY_LINK 0x00000400
- /* levels */
- #define HCI_SECURITY_AUTH 0x00001000
- #define HCI_SECURITY_ENCRYPT 0x00002000
- #define HCI_SECURITY_AUTHOR 0x00004000
-
- struct hci_dev_attr {
- int hd;
- char name[IFNAMSIZ];
- BD_ADDR bda;
- int flags;
- };
- struct hci_msg_hdr.
- #define MGR_STATE_CHANGE 0x01
-

- struct hci_msg_hdr {
- int opcode;
- int length;
- };
- struct hci_state_change.
- /* hci device events */
- #define HCIDEV_UP 0x0001
- #define HCIDEV_DOWN 0x0002
- #define HCIDEV_CHANGE 0x0004
- #define HCIDEV_REGISTER 0x0005
- #define HCIDEV_UNREGISTER 0x0006
- #define HCIDEV_ATTACH 0x0100
- #define HCIDEV_DETACH 0x0200
-
- struct hci_state_change {
- struct hci_msg_hdr hdr;
- int hd;
- int event;
- };

6.3.3.3 Παράδειγμα

Εδώ ακολουθεί ένα παράδειγμα της HCI API χρησιμότητας.

```
int main(int argc, char *argv[])
{
    int  devs[16], num, i, err;

    num = hci_get_devs(devs);
    if (num < 0) {
        printf("unable to get device list\n");
        return -1;
    }

    if (num == 0) {
        printf("No Bluetooth Adapters found\n");
        return 0;
    }
}
```



```

for (i = 0; i < num; i++) {
    fd = _hci_open_id(devs[i]);
    if (fd < 0) {
        printf("Unable to open Bluetooth device: %d\n\n", devs[i]);
        return -1;
    }

    err = hci_get_attr(fd, &da);
    if (err < 0) {
        printf("Unable to get attribute for: %d\n", devs[i]);
        return -1;
    }

    printf("%s\t%s\n", da.name, bda2str(&da.bda));

    if (!(da.state & HCI_STATE_UP)) {
        printf("\tDevice is down");
        continue;
    }

    err = HCI_WriteScanEnable(fd, 0x02);
    if (err) {
        printf("unable to set scan mode\n");
        return -1;
    }
}

```

Επόμενο παράδειγμα διευκρινίζει πώς να εντόπίσεις τις γύρω συσκευές.

```

int main(int argc, char *argv[])
{
    int    fd, i, argind = 1;
    __u32 length;
    int    err;
    INQUIRY_ITEM devs[20];
    char   *devnames[20];
    char   namebuf[248];
    __u8   num;

```

```

if (argv[argind]) {
    sscanf(argv[argind], "%x", &length);
} else
    length = 8;

fd = hci_open("bt0");
if (fd < 0) {
    printf("Unable to open device %s: %s\n", btdev, sys_errlist[errno]);
    return -1;
}

printf("Searching %d sec ...\n", length);
err = HCI_Inquiry(fd, length, 20, devs, &num);
print_hci_error(err);
if (num == 0) {
    printf("done.\nNo devices found.\n");
    return 0;
}
printf("Searching done. Resolving names ...\n");
for (i = 0; i < num; i++) {
    err = HCI_RemoteNameRequest(fd, &devs[i], namebuf);
    if (!err)
        devnames[i] = strdup(namebuf);
    else
        devnames[i] = NULL;
}
printf("done.\n");
for (i = 0; i < num; i++) {
    printf("%d, bda: %s, name: %s\n", i, bda2str(&devs[i].bda), devnames[i]);
    if (devnames[i])
        free(devnames[i]);
}
return 0;
}

```

6.3.4 SDP API

Το SDP API χωρίζεται σε δύο ζευγάρια: υπηρεσία πελάτη(client) API and υπηρεσία προμηθευτή (provider) API.

Η υπηρεσία client API είναι αφιερωμένη στους clients που βρίσκουν ποιες υπηρεσίες είναι διαθέσιμες στις απομακρυσμένες συσκευές, τι χαρακτηριστικά γνωρίσματα έχουν και πώς συνδέονται και χρησιμοποιούν κατάλληλα την υπηρεσία.

Ο προμηθευτής υπηρεσίας API είναι αφιερωμένος για υπηρεσίες στο να καταγράφουν πληροφορίες για τον εαυτό του με τον SDP server, επιτρέπει σε ένα client να τον βρει και να συνδεθεί μαζί του.

6.3.4.1 Χαρακτηριστικά γνωρίσματα των υπηρεσιών

Κάθε γνώρισμα υπηρεσίας περιγράφει ένα μοναδικό χαρακτηριστικό μιας υπηρεσίας.

Καθολικά χαρακτηριστικά ορίζονται στην προδιαγραφή του SDP. Ο ορισμός τους και ο σκοπός τους έχουν δηλωθεί πολύ καθαρά και είναι αρκετά κατανοητά, και επίσης περιγράφονται εν συντομία εδώ.

ServiceRecordHandle είναι μια μοναδική υπηρεσία identifier (32 bits) που ισχύει μόνο στη συσκευή που έχει τη διαχείριση. Η ίδια υπηρεσία σε δύο διαφορετικές συσκευές θα έχουν διαφορετική διαχείριση.

ServiceID είναι μια καθολική υπηρεσία identifier (UUID) που εγγυάται να είναι η ίδια σε όλες τις συσκευές όπου αυτή η υπηρεσία χρειάζεται.

ServiceClassIDSequence είναι η συνέχεια της UUIDs που αντιπροσωπεύει την ιεραρχία από τις κλάσεις της υπηρεσίας(type), η υπηρεσία είναι γνωστή για να συμμορφωθεί. Συνήθως ορίζεται με βάση "η πιο συγκεκριμένη" στη "πιο γενική".

AccessProtocols είναι μια ή περισσότερες ακολουθίες από descriptors πρωτοκόλλων. Ένας descriptor πρωτοκόλλων είναι μια ακολουθία από

χαρακτηριστικά – UUID των πρωτοκόλλων, αριθμός έκδοσης και αριθμός θύρας, από όπου έχουμε πρόσβαση στην υπηρεσία .

Bluetooth Profile Descriptor είναι μια ακολουθία από καθορισμένα "Bluetooth Profile" περιγραφές που η υπηρεσία είναι γνωστή για συμμόρφωση. Η Bluetooth περιγραφή των προφίλ περιέχει UUID του προφίλ, τον αριθμό της έκδοσης.

ServiceRecordState είναι ένας ακέραιος αριθμός is a 32-bit με την προϋπόθεση ότι θα διευκολύνει τις λειτουργίες των χαρακτηριστικών γνωρισμάτων των υπηρεσιών. Ένα αυτό το χαρακτηριστικό είναι παρών στο αρχείο υπηρεσιών, τότε εγγυάται να αλλάξει σε κάθε τροπολογία στο αρχείο.

ServiceInfoTimeToLive είναι ένας 32-bit ακέραιος που περιέχει τον αριθμό των δευτερολέπτων για όποιο αρχείο υπηρεσίας αναμένεται να μην αλλάξει, αλλά χωρίς αυτό να είναι εγγυημένο. Το χρονικό διάλειμμα μετράται από την στιγμή που το γνώρισμα φτάνει στον SDP server.

ServiceAvailability είναι ένας ακέραιος μη προσημασμένος 8 bit που αντιπροσωπεύει το κριτήριο (μεζούρα) της δυνατότητας της υπηρεσίας να διαχειρίζεται περισσότερους clients. 0x00 σημαίνει μηδενική διαθεσιμότητα και 0xff σημαίνει την μέγιστη διαθεσιμότητα..

ServiceName, *ServiceDescription* και *ServiceProviderName* είναι ορατές αλφαριθμητικές σειρές που άγουν πληροφορία σχετικά με την υπηρεσία. Μπορούν επίσης να υπάρχουν σε μια ή περισσότερες παγκόσμιες γλώσσες.

LanguageBasedAttributeIDList περιέχει συγκεκριμένους γλωσσικούς identifiers γνωρισμάτων για ορατά strings του χρήστη.

BrowseGroupList αποτελείται από μια συνέχεια από UUIDs που αντιπροσωπεύουν το "browse group" που ανήκει η υπηρεσία. Το Browse group είναι μια ιεραρχία από κατηγορίες υπηρεσιών, που διευκολύνουν την ανακάλυψη υπηρεσιών χωρίς μια "a priori" πληροφορία από κάθε υπηρεσία.

ClientExecutableURL τοποθεσία μιας πλατφόρμας ενός client (Win, Unix, Palm) σε συγκεκριμένη εφαρμογή που μπορεί να χρησιμοποιήσει την υπηρεσία.

DocumentationURL τοποθεσία από επιπρόσθετη τεκμηρίωση υπηρεσίας.

IconURL τοποθεσία μιας εικόνας που μπορεί να χρησιμοποιηθεί για να αναπαραστήσει μια υπηρεσία.

ServiceDatabaseState είναι ένα συγκεκριμένο γνώρισμα στην υπηρεσία που έχει ο server να ανακαλύπτει τον εαυτό του. Εάν η τιμή αλλάξει, τότε η αποθήκη της υπηρεσίας έχει πρόσφατα υποστεί τροποποιήσεις.

Επιπρόσθετα στα καθολικά χαρακτηριστικά, μια υπηρεσία μπορεί να έχει "service specific attributes". Αυτά τα χαρακτηριστικά βγάζουν νόημα μόνο στο περιβάλλον της υπηρεσίας. Δυναμικά χαρακτηριστικά μπορούν να είναι είτε βασικές είτε προκατασκευασμένες.

6.3.4.2 SDP δομές δεδομένων και γενικό API

6.3.4.2.1 UUID

Ένα UUID είναι ένας μοναδικός identifier που εγγυάται να είναι μοναδικός στον χώρο και στο χρόνο. UUIDs μπορούν ανεξάρτητα να δημιουργηθούν σε μια κατανεμημένη έκδοση. Μη κεντρική καταχώρηση των προσδιορισμένων UUIDs απαιτείται. Μια UUID είναι μια 128-bit τιμή.

Για να μειώσουμε το βάρος της αποθήκευσης και μετάδοσης 128-bit UUID τιμών, έχει προσδιοριστεί από πριν ένα εύρος από UUID τιμές για ανάθεση σε συχνά-χρησιμοποιούμενους καταγεγραμμένους σκοπούς. Η πρώτη UUID σε αυτό το προκαθορισμένο εύρος είναι γνωστή ως Bluetooth Base UUID και έχει την τιμή 00000000-0000-1000-8000- 00805F9B34FB, από την αναφορά Bluetooth Assigned Numbers . Οι UUID τιμές στο προκαθορισμένο εύρος έχουν ψευδώνυμο που αντιπροσωπεύουν 16-bit or 32-bit τιμές. Αυτά τα ψευδώνυμα συχνά καλούνται 16-bit και 32-bit UUIDs, αλλά είναι σημαντικό να σημειώσουμε ότι στην πραγματικότητα αντιπροσωπεύουν μια 128-bit UUID τιμή.

Το *UUID* αντικείμενο χρησιμοποιείται για να αποθηκεύσει uuid, το οποίο μπορεί να αποθηκεύσει 16, 32 ή 128 bits uuid. Για να απλοποιήσουμε τη διεπαφή σε αυτό το αντικείμενο, το *Affix* παρέχει μια ομάδα από συναρτήσεις για να δημιουργήσει μια τιμή και να τη μετατρέψει από έναν τύπο σε έναν άλλο.

Πίνακας 6.2.4.2.1. UUID API

| Συνάρτηση | Σκοπός |
|--|--|
| void makeUUIDFrom16Bits(UUID *pUUID, UUID16Bit value16Bit); | Δημιουργεί UUID object από μια 16 bit τιμή. |
| void makeUUIDFrom32Bits(UUID *pUUID, UUID32Bit value32Bit); | Δημιουργεί ένα UUID object από μια 32 bit τιμή. |
| void makeUUIDFrom128Bits(UUID *pUUID, UUID128Bit *value128Bit); | Δημιουργεί ένα UUID object από μια 128 bit τιμή. |
| UUID *convertTo128Bits(UUID *pUUID); | Μετατρέπει οποιονδήποτε τύπο UUID σε 128 bits UUID. Κατανέμει την μνήμη για να αποθηκεύσει ένα καινούργιο αντικείμενο. |

6.3.4.2.2 Καταγραφή υπηρεσιών

Όλη η πληροφορία για την υπηρεσία είναι ότι διατηρείται σε έναν SDP server μέσα σε έναν αρχείο-record υπηρεσίας. Αυτό το service record αποτελείται εξολοκλήρου από μια λίστα με χαρακτηριστικά των υπηρεσιών.

ServiceRecord object is used to store information about service. It maintains a list of all service attributes.

6.3.4.2.3 Δημιουργώντας χαρακτηριστικά γνωρίσματα

Η γενική δομή των δεδομένων *SDPData* δημιουργήθηκε για να κρατήσει οποιοδήποτε χαρακτηριστικό, δυναμικό ή καθολικό . Κάθε *SDPData* περιέχει ένα χαρακτηριστικό. Το *SDPData* αντικείμενο περιέχει τον identifier της υπηρεσίας χαρακτηριστικών, ένας identifier τύπου δεδομένων (int, char *, sequence etc), που ακολουθείται από μια τιμή.

Πίνακας 6.2.4.2.3 Γνώρισμα API

| Function | Purpose |
|---|--|
| SDPData *createSDPData(DataTypeDescriptor dtd, void *pValue); | Δημιουργεί ένα SDP attribute αντικείμενο ενός συγκεκριμένου τύπου. |
| Int addToAttributeList(ServiceRecord *svcRec, ServiceAttributeIdentifier attrId, SDPData *pSDPData, int replaceFlag); | Προσθέτει attribute αντικείμενο στην λίστα attribute list οαπό το ServiceRecord αντικείμενο. |
| void freeSDPData(SDPData *pData); | Καταστρέφει ένα ολόκληρο attribute αντικείμενο. |

6.3.4.3 Υπηρεσία πελάτη SDP API

Ο πελάτης υπηρεσίας SDP API χρησιμοποιείται από έναν client για να βρει υπηρεσίες που παρέχονται σε μια απομακρυσμένη συσκευή και να ανακαλύψει τι γνωρίσματα υπάρχουν σε αυτή. Όλες οι ομάδες συναρτήσεων παρουσιάζονται στον επόμενο πίνακα.:

Πίνακας 6.2.4.3 Υπηρεσία πελάτη SDP API

| Συνάρτηση | Σκοπός |
|---|---|
| int SDPInit(0); | Αρχικοποιεί την SDP υποδομή. |
| SDPServerHandle SDPOpenServerConnection(struct sockaddr_affix *saddr); | Ανοίγει μια σύνδεση με τον SDP server. |
| void SDPCloseServerConnection(SDPServerHandle srvHandle); | Κλείνει την σύνδεση με τον SDP server. |
| int SDPServiceSearchRequest(SDPServerHandle srvHandle, slist_t *svcSearchList, uint16_t maxSvcRecordCount, slist_t **svcResponseList, | Κάνει μια αίτηση αναζήτησης υπηρεσίας στον SDP server και |

| Συνάρτηση | Σκοπός |
|---|--|
| uint16_t *handleCountInResponse); | επιστρέφει μια υπηρεσία που έχει σε λίστα όλες τις υπηρεσίες που ταιριάζουν με το σχέδιο αυτής που έχει ζητηθεί. Σχέδιο αναζήτησης-λίστα από κλάση υπηρεσιών των identifiers. |
| int SDPServiceAttributeRequest(SDPServerHandle srvHandle, ServiceRecordHandle svcHandle, AttributeRequestType attrReqType, slist_t *attrIDList, uint16_t maxAttrIDByteCount, ServiceRecord **_svcRec, uint16_t *maxAttrResponseByteCount); | Κάνει μια αίτηση γνωρίσματος μιας υπηρεσίας που διαχειρίζεται "svcHandle" και επιστρέφει τη λίστα με τα γνωρίσματα που έχει ζητηθεί. Η "attrIDList" λίστα ορίζει γνωρίσματα να επιστρέψει. |
| int SDPServiceSearchAttributeRequest(SDPServerHandle srvHandle, slist_t *svcSearchList, AttributeRequestType attrReqType, slist_t *attrIDList, uint16_t maxAttrByteCount, slist_t **svcResponseList, uint16_t *maxAttrResponseByteCount); | Δημιουργεί μια υπηρεσία αίτησης αναζήτησης που επιστρέφει ορίσματα από όλες τις υπηρεσίες που βρέθηκαν. Συνδυάζει τις δύο προηγούμενες συναρτήσεις. |
| int __SDPServiceXXX(struct sockaddr_affix *saddr, ...); | Αυτές είναι ποικιλίες από τις τρεις προηγούμενες συναρτήσεις που παίρνουν την διεύθυνση για να στείλουν μια αίτηση αντί του |

| Συνάρτηση | Σκοπός |
|---|--|
| | ServerHandle. Ανοίγουν/ κλείνουν τη σύνδεση εσωτερικά. |
| int getXXX(ServiceRecord *svcRec,) | Περίγραμμα συναρτήσεων:εξάγει ανεξάρτητα ορίσματα από ένα αρχείο υπηρεσιών attributes (XXX) π.χ.: service name, access port, profile id, ... |

6.3.4.3.1 SDP client initialization

Η SDP υποδομή πρέπει να πάρει αρχικές τιμές πριν γίνει κάποια δραστηριότητα που να χρησιμοποιεί SDPInit(int mode).

6.3.4.3.2 Σύνδεση και αποσύνδεση με τον SDP server

Πριν γίνει οποιαδήποτε αίτηση στον SDP server, ένας client πρέπει να κάνει μια σύνδεση στον SDP server χρησιμοποιώντας την ακόλουθη συνάρτηση:

SDPServerHandle SDPOpenServerConnection(struct sockaddr_affix *saddr)

Αυτή η συνάρτηση δέχεται μια μόνο παράμετρο-διεύθυνσης του server και επιστρέφει SDP server handle.Μια διεύθυνση δίνεται χρησιμοποιώντας τη δομή του Affix sockaddr .

Για να κλείσεις μια σύνδεση ,πρέπει να καλέσεις την συνάρτηση, [void SDPCloseServerConnection(SDPServerHandle srvHandle);].

The ServerHandle παρέχει ένα πρώτο όρισμα στις SDPServiceXXX() συναρτήσεις.

6.3.4.3.3 Service class identifier(s) based search

Η υπηρεσία αναζήτησης δεδομένου ενός σχεδίου αναζήτησης αποτελείται κατά βάση από. Ο μέγιστος αριθμός των service class identifiers στο σχέδιο μπορεί να είναι 12. Η SDPServiceSearchRequest() χρησιμοποιείται για αυτό.

- ✦ Το σχέδιο αναζήτησης (svcSearchList) είναι μια λίστα από UUIDs (the UUID αντικείμενα που μπορούν να είναι είτε 16, 32 ή 128 bits). Τα UUID αντικείμενα μπορούν να δημιουργηθούν χρησιμοποιώντας βοηθητικές συναρτήσεις που ορίζονται στην τεχνική υποστήριξη του UUID.
- ✦ Ο client μπορεί να περιορίσει τον αριθμό των υπηρεσιών που διαχειρίζεται και περιμένει από έναν server, και έχει οριστεί στην in maxSvcRecordCount. Εάν δεν υπάρχει κάποιο όριο, το θέτει στη μεγαλύτερη τιμή.

Η απάντηση του server στην αίτηση είναι μια λίστα από αρχεία που ταιριάζουν ,και ένα setup στο svcResponseList (a slist_t) και μετά επιστρέφει σε αυτόν που είχε κάνει την κλήση. Ο αριθμός των handles που βρίσκονται είναι επίσης μια ομάδα στο handleCountInResponse.

6.3.4.3.4 Αίτηση υπηρεσίας χαρακτηριστικών από ένα συγκεκριμένο αρχείο καταγραφής.

Η αναζήτηση για ένα συγκεκριμένο set υπηρεσίας χαρακτηρίζεται στην συγκεκριμένη υπηρεσία. Αυτή η αίτηση συνήθως ακολουθεί έναν προηγούμενο service class identifier που βασίζεται στην αναζήτηση. Με το που έχει αναγνωριστεί η υπηρεσία, αυτή η αίτηση χρησιμοποιείται για να ανακαλύψει περισσότερα για την υπηρεσία. Ακολουθούν σύντομες περιγραφές των ορισμάτων :

- ✦ ServiceRecordHandle της υπηρεσίαςof χρειάζεται για να παρέχει απaráμιλλα την αναγνώριση στην υπηρεσία.
- ✦ Το API υποστηρίζει όλο το εύρος των γνωρισμάτων ή ένα συγκεκριμένο υποσύνολο και αυτό καθορίζεται στον τύπο αίτησης, και ακολουθείται

από μια λίστα με όλα τα γνωρίσματα , την attrIdList (slist_t). Οι identifiers γνωρισμάτων είναι 16 bit μη προσημασμένοι ακέραιοι που καθορίζονται με έναν από τους δύο τρόπους. *IndividualAttributes* - 16bit ιδιαίτεροι identifiers στην πραγματικότητα χαρακτηρίζονται identifiers με σειρά κατάταξη. *RangeOfAttributes* - 32bit identifier εύρος . Η υψηλή τάξη the 16bits είναι η αρχή του εύρους, η χαμηλή τάξη low-order 16bits είναι το τέλος του εύρους. Από 0x0000 σε 0xFFFF παίρνει όλα τα γνωρίσματα..

- ✦ maxAttrIDByteCount είναι η μέγιστη καταμέτρηση byte που ο πελάτης περιμένει να δεχτεί. Ο server δεν θα επιστρέψει ποτέ μια απάντηση που θα περιέχει attribute byte καταμέτρηση μεγαλύτερη από αυτή την τιμή.
- ✦ maxAttrByteCount – αυτός είναι ένας δείκτης σε έναν 16-bit ακέραιο, που έχει οριστεί να δείχνει τον αριθμό των bytes που επιστρέφει.
- ✦ Μια επιτυχημένη εκτέλεση της εντολής καταλήγει στην E_OK ως επιστροφή, διαφορετικά δείχνει σε έναν τύπο λάθους (timeout, invalid args etc).

6.3.4.3.5 Υπηρεσία class identifier(s) και υπηρεσία service attribute(s) based search

Αυτή η αναζήτηση συνδυάζει την υπηρεσία class identifier και την υπηρεσία αίτησης αναζήτησης γνωρισμάτων, έτσι ο αριθμός των αιτήσεων/ απαντήσεων συνδυασμών που χρειάζεται για να βρεθεί μια υπηρεσία γίνεται ο μικρότερος δυνατός. Οι παράμετροι περιέχουν ένα σχέδιο αναζήτησης υπηρεσίας και μια λίστα από γνωρίσματα που χρειάζονται για να εξαχθούν, και πρέπει ένα αρχείο υπηρεσίας να ταιριάζει με σχέδιο αίτησης αναζήτησης.

Για μια υπηρεσία που το ταίριασμα έχει γίνει και για την υπηρεσία που αποτελεί την αυτή που ψάχνουμε, τα ζητούμενα γνωρίσματα εξάγονται..

- ✦ The svcSearchList (slist_t) είναι μια μοναδικά συνδεδεμένη λίστα που χρησιμοποιεί στοιχεία ενός σχεδίου αναζήτησης.. Κάθε είσοδος στη λίστα είναι ένα UUID(DataTypeUUID_16) της υπηρεσίας που ψάχνουμε.

- ✦ AttributeSpecification attrSpec, ορίζει εάν η αίτηση είναι για ένα υποσύνολο γνωρισμάτων ή για όλο το εύρος που περιγράφεται στην Service attributes αίτηση από ένα συγκεκριμένο αρχείο.
- ✦ Το όρισμα attrIDList (slist_t) είναι ένα μοναδικά συνδεδεμένη λίστα που χρησιμοποιεί τους επιθυμητούς identifiers γνωρισμάτων. Κάθε στοιχείο είναι είτε ένας uint16_t(attrSpec = *IndividualAttributes*) είτε ένας uint32_t (attrSpec=*RangeOfAttributes*).
- ✦ Το όρισμα maxAttrIDByteCount είναι το byte καταμέτρησης του αριθμού του γνωρίσματος IDs που καθορίζεται σε μια λίστα αιτήσεων.
- ✦ Ο δείκτης svcResponseList (slist_t **) είναι ένα set σε μια επιτυχημένη επιστροφή ύστερα από μια αναζήτηση, και θα περιέχει το αρχείο υπηρεσίας που διαχειρίζεται της υπηρεσίες που ταιριάζουν.
- ✦ Το όρισμα maxAttrResponseByteCount είναι ένας δείκτης σε έναν 16-bit ακέραιο, ο οποίος έχει οριστεί να δείχνει τον αριθμό των bytes των γνωρισμάτων που επέστρεψαν. Αυτός ο δείκτης έχει οριστεί σε μία επιτυχή επιστροφή.

Εάν πριν την μέθοδο επιστρέψει E_OK σε μια επιτυχή εκτέλεση μιας αίτησης και ένας κώδικας λάθους αποτύχει..

6.3.4.3.6 SDP client συναρτήσεις χρησιμότητας

Οι πελάτες API στο προηγούμενο τομέα ήταν βασικοί και δεν παρείχαν μηχανισμούς για να εξάγουν γνωρίσματα από ατομικά ServiceRecord ή τα υποκομμάτια τους.

Οι getXXX(ServiceRecord *svcRec, XXX *xxx) συναρτήσεις που παρέχονται στην sdpclt.h βιβλιοθήκη, έχουν σκοπό να εξάγουν γνωρίσματα, όπου XXX είναι ένας τύπος γνωρίσματος. Αυτά δεν είναι στη λίστα εδώ, αλλά ο API ορισμός είναι εξηγείται μόνος του αυτόματα. Αυτά είναι συμπληρώματα των setXXX(XXX *x) συναρτήσεων χρησιμότητας για τον server, αλλά έχουν πακεταριστεί σε μια διαφορετική βιβλιοθήκη.

6.3.4.4 Προμηθευτής υπηρεσιών SDP API

Οι developers των Bluetooth υπηρεσιών μπορούν να χρησιμοποιήσουν το ακόλουθο API για να καταγράψουν υπηρεσίες με τον τοπικό SDP server. Οι "on-the-wire" διαμορφώσεις της καταγραφής υπηρεσιών μιμείται τον SDP. Ωστόσο μια διαφορά κλειδί είναι ότι η καταγραφή των υπηρεσιών δεν στέλνονται ποτέ στα κανάλια του Bluetooth, αλλά αυτό γίνεται χρησιμοποιώντας TCP sockets. Η καταγραφή υπηρεσιών αιτήσεις και απαντήσεις χρησιμοποιούν "δεσμευμένο εύρος" από SDP PDU Identifiers. Δεν δημιουργείται κάποιο πρόβλημα εφόσον δεν στέλνεται στα κανάλια του BT. Ο επόμενος πίνακας αποτελείται από τα APIs που είναι διαθέσιμα για τους υλοποιητές υπηρεσιών που καταγράφουν την υπηρεσία τους με τον SDP server πάνω στη συσκευή.

Πίνακας 6.2.4.4 Προμηθευτής υπηρεσιών SDP API

| Function | Purpose |
|--|---|
| int SDPInit(SDP_SVC_PROVIDER); | Αρχικοποιεί την υποδομή του SDP . |
| SDPServerHandle SDPOpenServerConnection(struct sockaddr_affix *saddr); | Ανοίγει μια σύνδεση με τον τοπικό SDP server. |
| Void SDPCloseServerConnection(SDPServerHandle srvHandle); | Κλείνει μια σύνδεση με τον SDP server. |
| ServiceRecord *createServiceRecord(void); | Creates a basic Service Record object w/o attributes. |
| void freeServiceRecord(ServiceRecord *svcRec); | Καταστρέφει ένα Service Record αντικείμενο και ελευθερώνει όλους τους διαθέσιμους πόρους. |
| ServiceRecord *createRFCOMMServiceRecord(int port, int flags); | Δημιουργεί ένα Service Record αντικείμενο και αποδίδει κοινά χαρακτηριστικά γνωρίσματα του RFCOMM . |

| Function | Purpose |
|---|--|
| int registerServiceRecord(SDPServerHandle srvHandle, ServiceRecord *svcRec); | Καταγράφει ένα Service Record αντικείμενο με τον SDP server. |
| int updateServiceRecord(SDPServerHandle srvHandle, ServiceRecord *svcRec); | Ενημερώνει συνεχώς ένα Service Record αντικείμενο στον the SDP server. |
| int deleteServiceRecord(SDPServerHandle srvHandle, ServiceRecord *svcRec); | Διαγράφει ένα Deletes a Service Record αντικείμενο από τον SDP server. |
| int setXXX(ServiceRecord *svcRec,) | Ένα περίγραμμα από συναρτήσεις : θέτει <<ατομικά>> χαρακτηριστικά (XXX) στο service record, e.g.: όνομα υπηρεσίας, θύρα πρόσβασης, προφίλ id, ... |

6.3.4.4.1 Αρχικοποίηση της SDP υποδομής

Η SDP υποδομή πρέπει να αρχικοποιηθεί πριν οποιαδήποτε δραστηριότητα και αυτό γίνεται χρησιμοποιώντας SDPInit(SDP_SVC_PROVIDER).

6.3.4.4.2 Σύνδεση και αποσύνδεση με τον SDP server

Πριν γίνει οποιαδήποτε καταγραφή υπηρεσιών ο προμηθευτής υπηρεσίας πρέπει να κάνει μια σύνδεση με τον τοπικό SDP server χρησιμοποιώντας την ακόλουθη συνάρτηση.

SDPServerHandle SDPOpenLocalServerConnection(void);

Η σύνδεση πρέπει να διατηρηθεί σε όλη τη διάρκεια ζωής της υπηρεσίας. Εάν για κάποιο λόγο η σύνδεση τερματιστεί τότε ο SDP server απομακρύνει όλες τις υπηρεσίες που ανήκουν σε αυτή την σύνδεση. Αυτό συμβαίνει ώστε να προληφθεί η περίπτωση όπου ο προμηθευτής υπηρεσιών τερματίζεται και δεν διαγράφει τις ήδη καταγεγραμμένες υπηρεσίες.

Για να τερματίσουμε μια σύνδεση, καλείται η `SDPCloseServerConnection()`.

6.3.4.4.3 Δημιουργία και τροποποίηση των χαρακτηριστικών γνωρισμάτων υπηρεσιών

Οι `setXXX(ServiceRecord *svcRec, XXX *value)` συναρτήσεις που παρέχονται στην βιβλιοθήκη `sdpsrv.h`, έχουν ως σκοπό το να δημιουργήσουν και να τροποποιήσουν τα χαρακτηριστικά γνωρίσματα υπηρεσιών, όπου το `XXX` είναι ένας attribute τύπος. Αυτές δεν εξηγούνται περαιτέρω εδώ, αλλά ο ορισμός του API είναι αυτό-επεξηγηματικός. Το μόνο σημείο που πρέπει να σημειώσουμε είναι ότι οποιοδήποτε παλιό `XXX` που παραμένει θα σβηστεί πριν θέσουμε την νέα τιμή.

Πρέπει επίσης να σημειωθεί ότι οι δείκτες που περνούν στις `setXXX()` συναρτήσεις δεν πρέπει να διαγραφούν για την διάρκεια ζωής της υπηρεσίας όπως αυτοί είναι αποθηκευμένοι. Δεν γίνονται αντίγραφα από χαρακτηριστικά γνωρίσματα που δείχνουν σε αυτούς τους δείκτες..

6.3.4.4.4 Διαγραφή του service record από τον SDP server

Διαγράφει ένα service record στο οποίο δείχνει ο `by svcRec`. Αυτό στην πραγματικότητα διαγράφει την απεικόνιση (image) του SDP server στο service record. Επιστρέφεται μηδέν 0 εάν η διαγραφή ήταν επιτυχημένη ή -1 εάν υπάρχει κάποια αποτυχία.

```
int deleteServiceRecord(ServiceRecord *svcRec);
```

Bluetooth ΠΗΓΗ1 <http://www.holtmann.org/linux/bluetooth/>

Bluetooth Πηγή 2 <http://affix.sourceforge.net/affix-doc/>

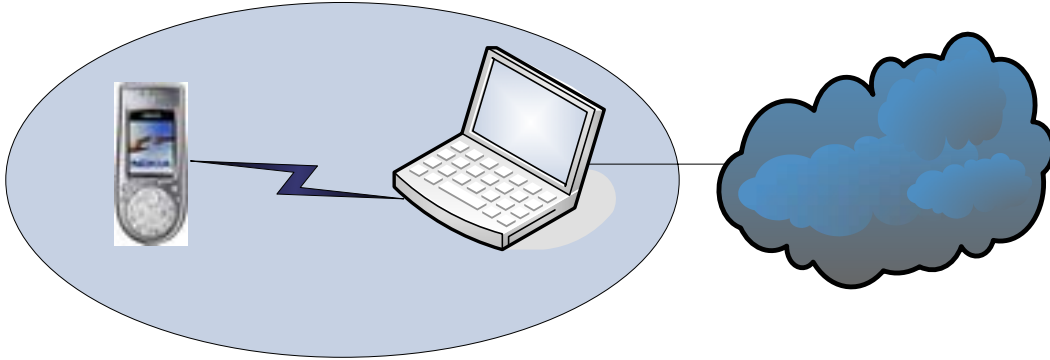
7 Εκτελεστική Διαδικασία

7.1 Σκοπός

Στο σημείο αυτό και αφού δώσαμε τις απαραίτητες θεωρητικές πληροφορίες, ήρθε η στιγμή να περιγράψουμε τη διπλωματική εργασία και ποιος ο σκοπός επίτευξής της. Σκοπός της διπλωματικής αυτής εργασίας είναι να καταφέρουμε να δώσουμε στο κινητό τηλέφωνο(στην περίπτωση μας NOKIA 3650-7650) ένα εικονικό ip ώστε να έχουμε πρόσβαση στο δίκτυο χωρίς τη χρήση GPRS υπηρεσιών. Έτσι πετυχαίνουμε ταχύτητες πλοήγησες αντίστοιχες με αυτές που έχουμε στους προσωπικούς υπολογιστές και κυρίως χωρίς χρέωση. Αυτό θα γίνει, έχοντας ένα υπολογιστή με λειτουργικό LINUX, να είναι συνδεδεμένος στο διαδίκτυο μέσω του LAN δικτύου του εργαστηρίου, του οποίου η ονομαστική ταχύτητα είναι τα 10Mbps. Στον υπολογιστή αυτό, θα έχουμε συνδεδεμένη μία USB Bluetooth συσκευή ώστε μέσω του Bluetooth να γίνει η σύνδεση υπολογιστή-κινητού τηλεφώνου. Το Dongle που χρησιμοποιήθηκε ήταν μάρκας TDK και είχε εμβέλεια μέχρι τα 150m. Στη συνέχεια, και αφού πετύχαμε την επιθυμητή σύνδεση του κινητού τηλεφώνου με τον υπολογιστή, προχωρήσαμε στο στάδιο να δώσουμε εικονικό ip στον κινητό τηλέφωνο και να πετύχουμε δωρεάν πλοήγηση στο διαδίκτυο. Στη συνέχεια με τη χρήση ενός Player πετύχαμε να δούμε streaming video μέσω του κινητού. Οι ταχύτητες πλοήγησης του κινητού στο διαδίκτυο είναι ισάξιες με αυτές που είναι συνδεδεμένες ο υπολογιστής. Στην περίπτωση μας είναι τα 10Mbps. Παρόλα αυτά, πετύχαμε streaming video μόνο με 64Kbps λόγω του Player ο οποίος δεν υποστήριζε μεγαλύτερες ταχύτητες. Η πρακτική χρησιμότητα μιας τέτοιας πλατφόρμας είναι μεγάλη και «ανοίγει» το δρόμο για άλλες υπηρεσίες τόσο σε «προσωπικό» επίπεδο όσο και σε εμπορικό. Τα τρία μεγαλύτερα προβλήματα που δεν έχουν λυθεί μέχρι τώρα είναι:

- ✦ Δεν υπάρχει Player που να υποστηρίζει ταχύτητες μεγαλύτερες των 64Kbps
- ✦ Τα κινητά τηλέφωνα δεν υποστηρίζουν master και slave ρόλους, ώστε να μπορούμε να συνδέσουμε πολλά κινητά τηλέφωνα σε έναν υπολογιστή. Δηλαδή ο υπολογιστής να λειτουργεί κατά κάποιο τρόπο σαν server

- ✦ Τρίτο και τελευταίο, οι μηνύες των κινητών τηλεφώνων, είναι αρκετά μικρές οπότε καμιά φορά κρεμάνε σε τέτοιου είδους εργασίες.



ΕΙΚΟΝΑ 7.1

7.2 Εισαγωγή

Για τους σκοπούς της παρούσας διπλωματικής εργασίας εγκαταστήσαμε το λειτουργικό σύστημα LINUX με έκδοση Mandrake 8.1. Κατόπιν προχωρήσαμε στην εγκατάσταση του Affix. Στο χρόνο που αυτή η εργασία έλαβε χώρα η νεότερη έκδοση του Affix ήταν η 2.0.2.

Επειδή η επίσημη στοίβα Bluetooth των LINUX είναι η Bluez είναι απαραίτητη η απεγκατάσταση του αφού αυτές οι δύο στοίβες δεν μπορούν να συνυπάρξουν μέσα στο ίδιο λειτουργικό.

Στο σημείο αυτό, πρέπει να σημειώσουμε, πως αφού όλα δουλέψανε σε τόσο παλαιές εκδόσεις LINUX, θα μπορούν όπως είναι φυσικό να δουλέψουν και σε μεταγενέστερες.

Στη συνέχεια εγκαταστήσαμε μία εφαρμογή στο κινητό, τη λεγόμενη GNUBOX η οποία είναι μια εφαρμογή γραμμένη σε Symbian. Παρακάτω, θα προσπαθήσουμε να δώσουμε όσο πιο αναλυτικά γίνεται και χωρίς περιττές πληροφορίες τον τρόπο για το «στήσιμο» της πλατφόρμας.

Για την εγκατάσταση του Affix πρέπει να γίνουν οι εξής διαδικασίες:

7.3 Προϋποθέσεις και οργάνωση πυρήνων

Εγκατάσταση των παρακάτω πακέτων:

libc - GNU C Βιβλιοθήκες και αρχεία επικεφαλίδων

pcmcia-cs - PCMCIA διαχειριστής συσκευών για PCMCIA Bluetooth συσκευές. Το πακέτο αυτό, χρειάζεται μόνο εάν είναι να χρησιμοποιηθεί PCMCIA Bluetooth προσαρμογέας. Σε αυτή τη διπλωματική, δεν χρησιμοποιήθηκε PCMCIA Bluetooth, οπότε και δεν ήταν απαραίτητη η εγκατάστασή του.

libopenobex-<version>-dev – Αρχεία ανάπτυξης πρωτοκόλλου OBEX.

libopenobex-<version> -Βιβλιοθήκες πρωτοκόλλου OBEX.

Εάν κάποιος επιθυμεί να έχει και affix pin gtk window χρειάζεται να εγκαταστήσει και τη python GTK library.

Το Affix μέχρι στιγμής υποστηρίζει 2.4.x και 2.6.x kernels(πυρήνες). Χρειάζεται να έχει γίνει πλήρης εγκατάσταση των εργαλείων μεταγλώττισης (compilers) των LINUX διότι το affix κατά την εγκατάστασή τους τα χρησιμοποιεί.

Επίσης συνιστάται να γίνει εγκατάσταση του πακέτου hotplug στην περίπτωση που θέλουμε να έχουμε υποστήριξη USB συσκευών.

Το Hotplug σου επιτρέπει να χρησιμοποιήσεις άμεσα μια συσκευή μόλις την προσαρτήσεις στον υπολογιστή σου. Αυτό σημαίνει ότι οι χρήστες δεν χρειάζεται να είναι γνώστες της διαχείρισης συσκευών από το σύστημα. Δηλαδή το σύστημα κάνει μόνο του εγκατάσταση της συσκευής. Αρχικά, το hotplug περιείχε υποστήριξη για USB και PCI συσκευές, και θα μπορούσε αυτόματα να σχηματίσει κάποια συνήθη interfaces δικτύου. Νεότερες εκδόσεις περιέχουν IEEE 1394 (Firewire/i.Link) υποστήριξη και μπορούν να λειτουργήσουν με σταθερότητα σε όποια USB συσκευή το χρειάζεται.

Η βασική έκδοση του hotplug περιλαμβάνεται σε όλες τις εκδόσεις GNU/Linux, συμπεριλαμβανομένων RedHat, Debian, και United Linux.

7.3.1 Εγκατάσταση και Μεταγλώττιση

Η εγκατάσταση του Affix απαιτεί την εγκατάσταση δύο διαφορετικών πακέτων: Του Affix και του Affix-Kernel.

Ο Affix-Kernel-<version> παρέχει τους απαραίτητους drivers και τις επικεφαλίδες που χρησιμοποιεί το Affix και το Affix-<version> από τη πλευρά του περιέχει χαρακτηριστικά για το affix bluetooth protocol stack.

Πρέπει να δώσουμε προσοχή στο ότι πρώτα εγκαθιστούμε τον Affix-Kernel και μετά το Affix.

7.3.2 Εγκατάσταση απαραίτητων πακέτων

Πριν ξεκινήσουμε την εγκατάσταση, πρέπει να βεβαιωθούμε πως έχουμε εισέλθει στον υπολογιστή σαν χρήστης root. Σε οποιαδήποτε άλλη περίπτωση, το σύστημα, δεν θα μας επιτρέψει να προχωρήσουμε στο στάδιο της εγκατάστασης.

Πριν προχωρήσουμε στην εγκατάσταση του affix δηλαδή του affix kernel και του affix, θα χρειαστεί να εγκαταστήσουμε κάποια πακέτα. Αυτά είναι το openobex και openobex-apps. Αφού πρώτα τα κατεβάσουμε από το δίκτυο, τα αποθηκεύουμε σε έναν φάκελο. Αφού βρισκόμαστε μέσα στο φάκελο, προχωράμε στην εγκατάσταση.

- ✱ Για το openobex, ακολουθούμε τα παρακάτω βήματα:
 - ./configure
 - make
 - make install

```
Konsole haris@Haris.Alexis /home Konsole
[haris@Haris haris]$ su
Password:
[root@Haris haris]# cd /home/haris/Diplwmtikh/
[root@Haris Diplwmtikh]# ls
affix-2.0.2/                hotplug-2003_08_05/
openobex-1.0.0/
affix-2.0.2.tar.gz*         hotplug-2003_08_05.tar.gz*
openobex-1.0.0.tar.gz*
affix-kernel-2.0.2/         libopenobex1-1[1].0.1-1mdk.i586.rpm*
openobex-apps-1.0.0/
affix-kernel-2.0.2.tar.gz*  libopenobex1-devel-1[1].0.1-
1mdk.i586.rpm* openobex-apps-1.0.0.tar.gz*
[root@Haris Diplwmtikh]# cd open
openobex-1.0.0              openobex-apps-1.0.0
```

```

openobex-1.0.0.tar.gz      openobex-apps-1.0.0.tar.gz
[root@Haris Diplwmtikh]# cd openobex-1
openobex-1.0.0             openobex-1.0.0.tar.gz
[root@Haris Diplwmtikh]# cd openobex-1.0.0
[root@Haris openobex-1.0.0]# ls
acinclude.m4      config.h.in      COPYING          m4macros/
makewindep.sh     openobex.spec.in
aclocal.m4        config.h.win32  doc/             Makefile.am
missing*          README
AUTHORS           config.sub*     INSTALL          Makefile.in
mkinstalldirs*    README.win32
ChangeLog         configure*      install-sh*      makefile.msc
NEWS              src/
config.guess*     configure.in     ltmain.sh        makefile.msc.rules
openobex-config.in stamp-h.in
[root@Haris openobex-1.0.0]# ./configure
creating cache ./config.cache
checking host system type... i686-pc-linux-gnu
checking target system type... i686-pc-linux-gnu
checking build system type... i686-pc-linux-gnu
checking host system type... i686-pc-linux-gnu
checking for a BSD compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking whether make sets ${MAKE}... yes
checking for working aclocal... found
checking for working autoconf... found
checking for working automake... found
checking for working autoheader... found
checking for working makeinfo... found
checking for gcc... gcc
checking whether the C compiler (gcc ) works... yes
checking whether the C compiler (gcc ) is a cross-compiler... no
checking whether we are using GNU C... yes
checking whether gcc accepts -g... yes
checking for a BSD compatible install... /usr/bin/install -c
checking for Cygwin environment... no
checking for mingw32 environment... no
checking how to run the C preprocessor... gcc -E
checking for ld used by GCC... /usr/bin/ld
checking if the linker (/usr/bin/ld) is GNU ld... yes
checking for /usr/bin/ld option to reload object files... -r
checking for BSD-compatible nm... /usr/bin/nm -B
checking whether ln -s works... yes
checking how to recognise dependant libraries... pass_all
checking for object suffix... o
checking for executable suffix... no
checking command to parse /usr/bin/nm -B output... ok
checking for dlfcn.h... yes
checking for ranlib... ranlib
checking for strip... strip
checking for objdir... .libs
checking for gcc option to produce PIC... -fPIC
checking if gcc PIC flag -fPIC works... yes
checking if gcc static flag -static works... yes
checking if gcc supports -c -o file.o... yes
checking if gcc supports -c -o file.lo... yes
checking if gcc supports -fno-rtti -fno-exceptions... yes
checking whether the linker (/usr/bin/ld) supports shared
libraries... yes
checking how to hardcode library paths into programs... immediate

```

```

checking whether stripping libraries is possible... yes
checking dynamic linker characteristics... GNU/Linux ld.so
checking if libtool supports shared libraries... yes
checking whether to build shared libraries... yes
checking whether to build static libraries... yes
checking whether -lc should be explicitly linked in... no
creating libtool
checking for IrDA support... yes
checking for Bluetooth support... no
updating cache ./config.cache
creating ./config.status
creating Makefile
creating m4macros/Makefile
creating src/Makefile
creating doc/Makefile
creating openobex-config
creating openobex.spec
creating config.h
[root@Haris openobex-1.0.0]# make
make all-recursive
make[1]: Entering directory `/home/haris/Diplwmtikh/openobex-1.0.0'
Making all in m4macros
make[2]: Entering directory `/home/haris/Diplwmtikh/openobex-1.0.0/m4macros'
make[2]: Nothing to be done for `all'.
make[2]: Leaving directory `/home/haris/Diplwmtikh/openobex-1.0.0/m4macros'
Making all in src
make[2]: Entering directory `/home/haris/Diplwmtikh/openobex-1.0.0/src'
/bin/sh ../libtool --mode=compile gcc -DHAVE_CONFIG_H -I. -I. -I.. -I../src -O2 -c obex.c
mkdir .libs
gcc -DHAVE_CONFIG_H -I. -I. -I.. -I../src -O2 -c obex.c -fPIC -DPIC -o .libs/obex.lo
In file included from obex_main.h:60,
                 from obex.c:49:
obex_transport.h:79:8: warning: extra tokens at end of #endif directive
gcc -DHAVE_CONFIG_H -I. -I. -I.. -I../src -O2 -c obex.c -o obex.o >/dev/null 2>&1
mv -f .libs/obex.lo obex.lo
/bin/sh ../libtool --mode=compile gcc -DHAVE_CONFIG_H -I. -I. -I.. -I../src -O2 -c obex_main.c
rm -f .libs/obex_main.lo
gcc -DHAVE_CONFIG_H -I. -I. -I.. -I../src -O2 -c obex_main.c -fPIC -DPIC -o .libs/obex_main.lo
In file included from obex_main.h:60,
                 from obex_main.c:54:
obex_transport.h:79:8: warning: extra tokens at end of #endif directive
gcc -DHAVE_CONFIG_H -I. -I. -I.. -I../src -O2 -c obex_main.c -o obex_main.o >/dev/null 2>&1
mv -f .libs/obex_main.lo obex_main.lo
/bin/sh ../libtool --mode=compile gcc -DHAVE_CONFIG_H -I. -I. -I.. -I../src -O2 -c obex_connect.c
rm -f .libs/obex_connect.lo
gcc -DHAVE_CONFIG_H -I. -I. -I.. -I../src -O2 -c obex_connect.c -fPIC -DPIC -o .libs/obex_connect.lo

```

```

In file included from obex_main.h:60,
                    from obex_connect.c:38:
obex_transport.h:79:8: warning: extra tokens at end of #endif
directive
gcc -DHAVE_CONFIG_H -I. -I. -I.. -I../src -O2 -c obex_connect.c -
-o obex_connect.o >/dev/null 2>&1
mv -f .libs/obex_connect.lo obex_connect.lo
/bin/sh ../libtool --mode=compile gcc -DHAVE_CONFIG_H -I. -I. -
I.. -I../src -O2 -c obex_header.c
rm -f .libs/obex_header.lo
gcc -DHAVE_CONFIG_H -I. -I. -I.. -I../src -O2 -c obex_header.c -
fPIC -DPIC -o .libs/obex_header.lo
In file included from obex_main.h:60,
                    from obex_header.c:38:
obex_transport.h:79:8: warning: extra tokens at end of #endif
directive
gcc -DHAVE_CONFIG_H -I. -I. -I.. -I../src -O2 -c obex_header.c -o
obex_header.o >/dev/null 2>&1
mv -f .libs/obex_header.lo obex_header.lo
/bin/sh ../libtool --mode=compile gcc -DHAVE_CONFIG_H -I. -I. -
I.. -I../src -O2 -c obex_object.c
rm -f .libs/obex_object.lo
gcc -DHAVE_CONFIG_H -I. -I. -I.. -I../src -O2 -c obex_object.c -
fPIC -DPIC -o .libs/obex_object.lo
In file included from obex_main.h:60,
                    from obex_object.c:38:
obex_transport.h:79:8: warning: extra tokens at end of #endif
directive
gcc -DHAVE_CONFIG_H -I. -I. -I.. -I../src -O2 -c obex_object.c -o
obex_object.o >/dev/null 2>&1
mv -f .libs/obex_object.lo obex_object.lo
/bin/sh ../libtool --mode=compile gcc -DHAVE_CONFIG_H -I. -I. -
I.. -I../src -O2 -c obex_transport.c
rm -f .libs/obex_transport.lo
gcc -DHAVE_CONFIG_H -I. -I. -I.. -I../src -O2 -c obex_transport.c
-fPIC -DPIC -o .libs/obex_transport.lo
In file included from obex_main.h:60,
                    from obex_transport.c:38:
obex_transport.h:79:8: warning: extra tokens at end of #endif
directive
gcc -DHAVE_CONFIG_H -I. -I. -I.. -I../src -O2 -c obex_transport.c
-o obex_transport.o >/dev/null 2>&1
mv -f .libs/obex_transport.lo obex_transport.lo
/bin/sh ../libtool --mode=compile gcc -DHAVE_CONFIG_H -I. -I. -
I.. -I../src -O2 -c obex_server.c
rm -f .libs/obex_server.lo
gcc -DHAVE_CONFIG_H -I. -I. -I.. -I../src -O2 -c obex_server.c -
fPIC -DPIC -o .libs/obex_server.lo
In file included from obex_main.h:60,
                    from obex_server.c:38:
obex_transport.h:79:8: warning: extra tokens at end of #endif
directive
gcc -DHAVE_CONFIG_H -I. -I. -I.. -I../src -O2 -c obex_server.c -o
obex_server.o >/dev/null 2>&1
mv -f .libs/obex_server.lo obex_server.lo
/bin/sh ../libtool --mode=compile gcc -DHAVE_CONFIG_H -I. -I. -
I.. -I../src -O2 -c obex_client.c
rm -f .libs/obex_client.lo
gcc -DHAVE_CONFIG_H -I. -I. -I.. -I../src -O2 -c obex_client.c -
fPIC -DPIC -o .libs/obex_client.lo

```

```

In file included from obex_main.h:60,
                    from obex_client.c:37:
obex_transport.h:79:8: warning: extra tokens at end of #endif
directive
gcc -DHAVE_CONFIG_H -I. -I. -I.. -I../src -O2 -c obex_client.c -o
obex_client.o >/dev/null 2>&1
mv -f .libs/obex_client.lo obex_client.lo
/bin/sh ../libtool --mode=compile gcc -DHAVE_CONFIG_H -I. -I. -
I.. -I../src -O2 -c irobex.c
rm -f .libs/irobex.lo
gcc -DHAVE_CONFIG_H -I. -I. -I.. -I../src -O2 -c irobex.c -fPIC
-DPIC -o .libs/irobex.lo
In file included from obex_main.h:60,
                    from irobex.c:60:
obex_transport.h:79:8: warning: extra tokens at end of #endif
directive
gcc -DHAVE_CONFIG_H -I. -I. -I.. -I../src -O2 -c irobex.c -o
irobex.o >/dev/null 2>&1
mv -f .libs/irobex.lo irobex.lo
/bin/sh ../libtool --mode=compile gcc -DHAVE_CONFIG_H -I. -I. -
I.. -I../src -O2 -c inobex.c
rm -f .libs/inobex.lo
gcc -DHAVE_CONFIG_H -I. -I. -I.. -I../src -O2 -c inobex.c -fPIC
-DPIC -o .libs/inobex.lo
In file included from obex_main.h:60,
                    from inobex.c:45:
obex_transport.h:79:8: warning: extra tokens at end of #endif
directive
gcc -DHAVE_CONFIG_H -I. -I. -I.. -I../src -O2 -c inobex.c -o
inobex.o >/dev/null 2>&1
mv -f .libs/inobex.lo inobex.lo
/bin/sh ../libtool --mode=compile gcc -DHAVE_CONFIG_H -I. -I. -
I.. -I../src -O2 -c btobex.c
rm -f .libs/btobex.lo
gcc -DHAVE_CONFIG_H -I. -I. -I.. -I../src -O2 -c btobex.c -fPIC
-DPIC -o .libs/btobex.lo
gcc -DHAVE_CONFIG_H -I. -I. -I.. -I../src -O2 -c btobex.c -o
btobex.o >/dev/null 2>&1
mv -f .libs/btobex.lo btobex.lo
/bin/sh ../libtool --mode=compile gcc -DHAVE_CONFIG_H -I. -I. -
I.. -I../src -O2 -c netbuf.c
rm -f .libs/netbuf.lo
gcc -DHAVE_CONFIG_H -I. -I. -I.. -I../src -O2 -c netbuf.c -fPIC
-DPIC -o .libs/netbuf.lo
gcc -DHAVE_CONFIG_H -I. -I. -I.. -I../src -O2 -c netbuf.c -o
netbuf.o >/dev/null 2>&1
mv -f .libs/netbuf.lo netbuf.lo
/bin/sh ../libtool --mode=link gcc -O2 -o libopenobex.la -rpath
/usr/local/lib -version-info 0:0:0 -release 1.0 -export-
symbols ../src/obex.sym obex.lo obex_main.lo obex_connect.lo
obex_header.lo obex_object.lo obex_transport.lo obex_server.lo
obex_client.lo irobex.lo inobex.lo btobex.lo netbuf.lo
rm -fr .libs/libopenobex.la .libs/libopenobex.*
.libs/libopenobex-1.0.*
gcc -shared obex.lo obex_main.lo obex_connect.lo obex_header.lo
obex_object.lo obex_transport.lo obex_server.lo obex_client.lo
irobex.lo inobex.lo btobex.lo netbuf.lo -Wl,-soname -
Wl,libopenobex-1.0.so.0 -Wl,-retain-symbols-file -
Wl,../src/obex.sym -o .libs/libopenobex-1.0.so.0.0.0
(cd .libs && rm -f libopenobex-1.0.so.0 && ln -s libopenobex-

```

```

1.0.so.0.0.0 libopenobex-1.0.so.0)
(cd .libs && rm -f libopenobex.so && ln -s libopenobex-
1.0.so.0.0.0 libopenobex.so)
ar cru .libs/libopenobex.a obex.o obex_main.o obex_connect.o
obex_header.o obex_object.o obex_transport.o obex_server.o
obex_client.o irobex.o inobex.o btobex.o netbuf.o
ranlib .libs/libopenobex.a
creating libopenobex.la
(cd .libs && rm -f libopenobex.la && ln -s ../libopenobex.la
libopenobex.la)
make[2]: Leaving directory `/home/haris/Diplwmtikh/openobex-
1.0.0/src'
Making all in doc
make[2]: Entering directory `/home/haris/Diplwmtikh/openobex-
1.0.0/doc'
gcc -DHAVE_CONFIG_H -I. -I. -I.. -O2 -c docproc.c
/bin/sh ../libtool --mode=link gcc -O2 -o docproc docproc.o
mkdir .libs
gcc -O2 -o docproc docproc.o
make[2]: Leaving directory `/home/haris/Diplwmtikh/openobex-
1.0.0/doc'
make[2]: Entering directory `/home/haris/Diplwmtikh/openobex-
1.0.0'
make[2]: Nothing to be done for `all-am'.
make[2]: Leaving directory `/home/haris/Diplwmtikh/openobex-
1.0.0'
make[1]: Leaving directory `/home/haris/Diplwmtikh/openobex-
1.0.0'
[root@Haris openobex-1.0.0]# make install
Making install in m4macros
make[1]: Entering directory `/home/haris/Diplwmtikh/openobex-
1.0.0/m4macros'
make[2]: Entering directory `/home/haris/Diplwmtikh/openobex-
1.0.0/m4macros'
make[2]: Nothing to be done for `install-exec-am'.
/bin/sh ../mkinstalldirs /usr/local/share/aclocal
mkdir /usr/local/share/aclocal
/usr/bin/install -c -m 644 ./openobex.m4
/usr/local/share/aclocal/openobex.m4
make[2]: Leaving directory `/home/haris/Diplwmtikh/openobex-
1.0.0/m4macros'
make[1]: Leaving directory `/home/haris/Diplwmtikh/openobex-
1.0.0/m4macros'
Making install in src
make[1]: Entering directory `/home/haris/Diplwmtikh/openobex-
1.0.0/src'
make[2]: Entering directory `/home/haris/Diplwmtikh/openobex-
1.0.0/src'
/bin/sh ../mkinstalldirs /usr/local/lib
/bin/sh ../libtool --mode=install /usr/bin/install -c
libopenobex.la /usr/local/lib/libopenobex.la
/usr/bin/install -c .libs/libopenobex-1.0.so.0.0.0
/usr/local/lib/libopenobex-1.0.so.0.0.0
(cd /usr/local/lib && rm -f libopenobex-1.0.so.0 && ln -s
libopenobex-1.0.so.0.0.0 libopenobex-1.0.so.0)
(cd /usr/local/lib && rm -f libopenobex.so && ln -s libopenobex-
1.0.so.0.0.0 libopenobex.so)
/usr/bin/install -c .libs/libopenobex.lai
/usr/local/lib/libopenobex.la
/usr/bin/install -c .libs/libopenobex.a

```



```

/usr/local/lib/libopenobex.a
ranlib /usr/local/lib/libopenobex.a
chmod 644 /usr/local/lib/libopenobex.a
PATH="$PATH:/sbin" ldconfig -n /usr/local/lib
-----
-----
Libraries have been installed in:
    /usr/local/lib

If you ever happen to want to link against installed libraries
in a given directory, LIBDIR, you must either use libtool, and
specify the full pathname of the library, or use the '-LLIBDIR'
flag during linking and do at least one of the following:
    - add LIBDIR to the 'LD_LIBRARY_PATH' environment variable
      during execution
    - add LIBDIR to the 'LD_RUN_PATH' environment variable
      during linking
    - use the '-Wl,--rpath -Wl,LIBDIR' linker flag
    - have your system administrator add LIBDIR to
      '/etc/ld.so.conf'

See any operating system documentation about shared libraries for
more information, such as the ld(1) and ld.so(8) manual pages.
-----
-----
/bin/sh ../mkinstalldirs /usr/local/include/openobex
mkdir /usr/local/include/openobex
/usr/bin/install -c -m 644 obex.h
/usr/local/include/openobex/obex.h
/usr/bin/install -c -m 644 obex_const.h
/usr/local/include/openobex/obex_const.h
make[2]: Leaving directory `/home/haris/Diplwmtikh/openobex-
1.0.0/src'
make[1]: Leaving directory `/home/haris/Diplwmtikh/openobex-
1.0.0/src'
Making install in doc
make[1]: Entering directory `/home/haris/Diplwmtikh/openobex-
1.0.0/doc'
make[2]: Entering directory `/home/haris/Diplwmtikh/openobex-
1.0.0/doc'
make[2]: Nothing to be done for `install-exec-am'.
make[2]: Nothing to be done for `install-data-am'.
make[2]: Leaving directory `/home/haris/Diplwmtikh/openobex-
1.0.0/doc'
make[1]: Leaving directory `/home/haris/Diplwmtikh/openobex-
1.0.0/doc'
make[1]: Entering directory `/home/haris/Diplwmtikh/openobex-
1.0.0'
make[2]: Entering directory `/home/haris/Diplwmtikh/openobex-
1.0.0'
/bin/sh ../mkinstalldirs /usr/local/bin
/usr/bin/install -c openobex-config /usr/local/bin/openobex-
config
make[2]: Nothing to be done for `install-data-am'.
make[2]: Leaving directory `/home/haris/Diplwmtikh/openobex-
1.0.0'
make[1]: Leaving directory `/home/haris/Diplwmtikh/openobex-
1.0.0'
[root@Haris openobex-1.0.0]#

```

Ενδέχεται να λείπουν από το σύστημα κάποια rpm αρχεία. Αυτά θα ζητηθούν από το σύστημα και το μόνο που έχετε να κάνετε είναι να τα κατεβάσετε από το δίκτυο και να τα εγκαταστήσετε πριν αρχίσετε την κυρίως εγκατάσταση.

- ✦ Αφού ολοκληρώσατε την εγκατάσταση του openobex, προχωράμε στην εγκατάσταση του openobex-apps που γίνεται σχεδόν με τον ίδιο τρόπο
 - ./configure
 - make
 - make install

```
Konsole haris@Haris.Alexis /home Konsole
[root@Haris Diplwmtikh]# cd openobex-apps-1.0.0
[root@Haris openobex-apps-1.0.0]# ./configure
creating cache ./config.cache
checking for a BSD compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking whether make sets ${MAKE}... yes
checking for working aclocal... found
checking for working autoconf... found
checking for working automake... found
checking for working autoheader... found
checking for working makeinfo... found
checking whether to enable maintainer-specific portions of
Makefiles... no
checking for gcc... gcc
checking whether the C compiler (gcc ) works... yes
checking whether the C compiler (gcc ) is a cross-compiler... no
checking whether we are using GNU C... yes
checking whether gcc accepts -g... yes
checking for a BSD compatible install... /usr/bin/install -c
checking for ranlib... ranlib
checking for openobex-config... /usr/bin/openobex-config
checking for openobex - version >= 1.0.0... yes
checking for Bluetooth support... no
updating cache ./config.cache
creating ./config.status
creating Makefile
creating src/Makefile
creating config.h
[root@Haris openobex-apps-1.0.0]# make
make all-recursive
make[1]: Entering directory `/home/haris/Diplwmtikh/openobex-
apps-1.0.0'
Making all in src
make[2]: Entering directory `/home/haris/Diplwmtikh/openobex-
apps-1.0.0/src'
gcc -DHAVE_CONFIG_H -I. -I. -I.. -I/usr/include      -g -O2 -c
obex_io.c
gcc -DHAVE_CONFIG_H -I. -I. -I.. -I/usr/include      -g -O2 -c
obex_put_common.c
rm -f libmisc.a
ar cru libmisc.a obex_io.o obex_put_common.o
ranlib libmisc.a
gcc -DHAVE_CONFIG_H -I. -I. -I.. -I/usr/include      -g -O2 -c
irxfer.c
gcc -g -O2 -o irxfer irxfer.o -lopenobex libmisc.a
gcc -DHAVE_CONFIG_H -I. -I. -I.. -I/usr/include      -g -O2 -c
```

```

obex_tcp.c
gcc -g -O2 -o obex_tcp obex_tcp.o -lopenobex libmisc.a
gcc -DHAVE_CONFIG_H -I. -I. -I.. -I/usr/include -g -O2 -c
irobex_palm3.c
gcc -g -O2 -o irobex_palm3 irobex_palm3.o -lopenobex libmisc.a
gcc -DHAVE_CONFIG_H -I. -I. -I.. -I/usr/include -g -O2 -c
obex_test.c
gcc -DHAVE_CONFIG_H -I. -I. -I.. -I/usr/include -g -O2 -c
obex_test_client.c
gcc -DHAVE_CONFIG_H -I. -I. -I.. -I/usr/include -g -O2 -c
obex_test_server.c
gcc -DHAVE_CONFIG_H -I. -I. -I.. -I/usr/include -g -O2 -c
obex_test_cable.c
gcc -g -O2 -o obex_test obex_test.o obex_test_client.o
obex_test_server.o obex_test_cable.o -lopenobex libmisc.a
make[2]: Leaving directory `/home/haris/Diplwmtikh/openobex-apps-
1.0.0/src'
make[2]: Entering directory `/home/haris/Diplwmtikh/openobex-
apps-1.0.0'
make[2]: Nothing to be done for `all-am'.
make[2]: Leaving directory `/home/haris/Diplwmtikh/openobex-apps-
1.0.0'
make[1]: Leaving directory `/home/haris/Diplwmtikh/openobex-apps-
1.0.0'
[root@Haris openobex-apps-1.0.0]# make install
Making install in src
make[1]: Entering directory `/home/haris/Diplwmtikh/openobex-
apps-1.0.0/src'
make[2]: Entering directory `/home/haris/Diplwmtikh/openobex-
apps-1.0.0/src'
/bin/sh ../mkinstalldirs /usr/local/bin
  /usr/bin/install -c irxfer /usr/local/bin/irxfer
  /usr/bin/install -c obex_tcp /usr/local/bin/obex_tcp
  /usr/bin/install -c irobex_palm3 /usr/local/bin/irobex_palm3
  /usr/bin/install -c obex_test /usr/local/bin/obex_test
make[2]: Nothing to be done for `install-data-am'.
make[2]: Leaving directory `/home/haris/Diplwmtikh/openobex-apps-
1.0.0/src'
make[1]: Leaving directory `/home/haris/Diplwmtikh/openobex-apps-
1.0.0/src'
make[1]: Entering directory `/home/haris/Diplwmtikh/openobex-
apps-1.0.0'
make[2]: Entering directory `/home/haris/Diplwmtikh/openobex-
apps-1.0.0'
make[2]: Nothing to be done for `install-exec-am'.
make[2]: Nothing to be done for `install-data-am'.
make[2]: Leaving directory `/home/haris/Diplwmtikh/openobex-apps-
1.0.0'
make[1]: Leaving directory `/home/haris/Diplwmtikh/openobex-apps-
1.0.0'
[root@Haris openobex-apps-1.0.0]#

```

Αφού ολοκληρώσατε τα παραπάνω με επιτυχία, μπορείτε να προχωρήσετε στο επόμενο βήμα που βρίσκεται στην επόμενη ακριβώς παράγραφο.

7.3.3 Εγκατάσταση του Affix Kernel

Κατεβάζουμε τα απαραίτητα αρχεία, τα οποία είναι: Affix και Affix-Kernel.

Αφού έχουμε κάνει login στο μηχάνημα σαν root και αφού βρισκόμαστε στο μονοπάτι-φάκελο που έχουμε τοποθετήσει το πακέτο Affix-Kernel προχωράμε στη διαδικασία της εγκατάστασης:

1. Το πρώτο βήμα, περιλαμβάνει το configure του αρχείου
Το οποίο γίνεται πληκτρολογώντας την εντολή:

➤ **make config**

στο directory που βρίσκεται το Affix-Kernel. Εάν δεν θέλουμε να κάνουμε debugging, λέμε No στην αντίστοιχη επιλογή. (Συνιστούμε να μην γίνει Debugging).

2. Προχωράμε στο στάδιο του Compiling
Πληκτρολογούμε την εντολή:

➤ **make all**

3. Τελευταίο στάδιο είναι το στάδιο της εγκατάστασης.
Αυτό γίνεται με την εντολή:

➤ **make install**

Konsole haris@Haris.Alexis /home Konsole

```
[root@Haris haris]# cd /home/haris/Diplwmtikh/affix-kernel-2.0.2
[root@Haris affix-kernel-2.0.2]# make config
Cleaning programs
```

```
----- Affix Configuration Script -----
```

```
Linux source directory [/usr/src/linux]:
```

```

The kernel source tree is version 2.4.8-26mdk.

Target architecture (i386,arm,ppc,...) [i386]:

Module install directory [/lib/modules/2.4.8-26mdk]:

    Common Section

Debug enabled (y/n) [n]: n

    Protocol Section

SCO support (Audio) (y/n) [y]: y
L2CAP support (y/n) [y]: y
RFCOMM support (y/n) [y]: y
PAN support (y/n) [y]: y

    Drivers Section

USB device support (y/n) [y]: y
UART device support (required for PCMCIA UART) (y/n) [y]: n
PCMCIA Bluecard device support (Anycom) (y/n) [y]: n
PCMCIA 3COM device support (y/n) [y]: n
PCMCIA bt950 device support (Ambicom/Pretec CF cards) (y/n) [y]:
n

Configuration successful.

[root@Haris affix-kernel-2.0.2]# make all
Checking kernel checksum ...
make[1]: Entering directory `/home/haris/Diplwmtikh/affix-kernel-
2.0.2'
Compiling driver and control programs
make[2]: Entering directory `/home/haris/Diplwmtikh/affix-kernel-
2.0.2/etc'
make[2]: Nothing to be done for `all'.
make[2]: Leaving directory `/home/haris/Diplwmtikh/affix-kernel-
2.0.2/etc'
make[2]: Entering directory `/home/haris/Diplwmtikh/affix-kernel-
2.0.2/include'
make[2]: Leaving directory `/home/haris/Diplwmtikh/affix-kernel-
2.0.2/include'
make[2]: Entering directory `/home/haris/Diplwmtikh/affix-kernel-
2.0.2/btcore'
gcc -MD -O2 -Wall -Wstrict-prototypes -fomit-frame-pointer -pipe
-I/home/haris/Diplwmtikh/affix-kernel-2.0.2/include -
I/usr/src/linux/include -include /home/haris/Diplwmtikh/affix-
kernel-2.0.2/include/affix.h -D__KERNEL__ -DMODULE -DMODVERSIONS
-include /usr/src/linux/include/linux/modversions.h -
DEXPORT_SYMTAB -c hci.c
gcc -MD -O2 -Wall -Wstrict-prototypes -fomit-frame-pointer -pipe
-I/home/haris/Diplwmtikh/affix-kernel-2.0.2/include -
I/usr/src/linux/include -include /home/haris/Diplwmtikh/affix-
kernel-2.0.2/include/affix.h -D__KERNEL__ -DMODULE -DMODVERSIONS
-include /usr/src/linux/include/linux/modversions.h -
DEXPORT_SYMTAB -c af_hci.c
gcc -MD -O2 -Wall -Wstrict-prototypes -fomit-frame-pointer -pipe
-I/home/haris/Diplwmtikh/affix-kernel-2.0.2/include -

```

```

I/usr/src/linux/include -include /home/haris/Diplwmtikh/affix-
kernel-2.0.2/include/affix.h -D__KERNEL__ -DMODULE -DMODVERSIONS
-include /usr/src/linux/include/linux/modversions.h -
DEXPORT_SYMTAB -c l2cap.c
/home/haris/Diplwmtikh/affix-kernel-2.0.2/include/affix/hci.h:67:
warning: `sprintf_chk' defined but not used gcc -MD -O2 -Wall -
Wstrict-prototypes -fomit-frame-pointer -pipe -
I/home/haris/Diplwmtikh/affix-kernel-2.0.2/include -
I/usr/src/linux/include -include /home/haris/Diplwmtikh/affix-
kernel-2.0.2/include/affix.h -D__KERNEL__ -DMODULE -DMODVERSIONS
-include /usr/src/linux/include/linux/modversions.h -c
hci_sched.c
/home/haris/Diplwmtikh/affix-kernel-2.0.2/include/affix/hci.h:67:
warning: `sprintf_chk' defined but not used gcc -MD -O2 -Wall -
Wstrict-prototypes -fomit-frame-pointer -pipe -
I/home/haris/Diplwmtikh/affix-kernel-2.0.2/include -
I/usr/src/linux/include -include /home/haris/Diplwmtikh/affix-
kernel-2.0.2/include/affix.h -D__KERNEL__ -DMODULE -DMODVERSIONS
-include /usr/src/linux/include/linux/modversions.h -c hci_mgr.c
/home/haris/Diplwmtikh/affix-kernel-2.0.2/include/affix/hci.h:67:
warning: `sprintf_chk' defined but not used gcc -MD -O2 -Wall -
Wstrict-prototypes -fomit-frame-pointer -pipe -
I/home/haris/Diplwmtikh/affix-kernel-2.0.2/include -
I/usr/src/linux/include -include /home/haris/Diplwmtikh/affix-
kernel-2.0.2/include/affix.h -D__KERNEL__ -DMODULE -DMODVERSIONS
-include /usr/src/linux/include/linux/modversions.h -c sysctl-
core.c
/home/haris/Diplwmtikh/affix-kernel-2.0.2/include/affix/hci.h:67:
warning: `sprintf_chk' defined but not used sysctl-core.c:81:
warning: `affix_core_table' defined but not used
gcc -MD -O2 -Wall -Wstrict-prototypes -fomit-frame-pointer -pipe
-I/home/haris/Diplwmtikh/affix-kernel-2.0.2/include -
I/usr/src/linux/include -include /home/haris/Diplwmtikh/affix-
kernel-2.0.2/include/affix.h -D__KERNEL__ -DMODULE -DMODVERSIONS
-include /usr/src/linux/include/linux/modversions.h -c af_l2cap.c
/home/haris/Diplwmtikh/affix-kernel-2.0.2/include/affix/hci.h:67:
warning: `sprintf_chk' defined but not used ld -r -o
../bin/affix.o hci_sched.o hci_mgr.o sysctl-core.o af_l2cap.o
hci.o af_hci.o l2cap.o
chmod -x ../bin/affix.o
make[2]: Leaving directory `/home/haris/Diplwmtikh/affix-kernel-
2.0.2/btcore'
make[2]: Entering directory `/home/haris/Diplwmtikh/affix-kernel-
2.0.2/rfcomm'
gcc -MD -O2 -Wall -Wstrict-prototypes -fomit-frame-pointer -pipe
-I/home/haris/Diplwmtikh/affix-kernel-2.0.2/include -
I/usr/src/linux/include -include /home/haris/Diplwmtikh/affix-
kernel-2.0.2/include/affix.h -D__KERNEL__ -DMODULE -DMODVERSIONS
-include /usr/src/linux/include/linux/modversions.h -c rfcomm.c
/home/haris/Diplwmtikh/affix-kernel-2.0.2/include/affix/hci.h:67:
warning: `sprintf_chk' defined but not used gcc -MD -O2 -Wall -
Wstrict-prototypes -fomit-frame-pointer -pipe -
I/home/haris/Diplwmtikh/affix-kernel-2.0.2/include -
I/usr/src/linux/include -include /home/haris/Diplwmtikh/affix-
kernel-2.0.2/include/affix.h -D__KERNEL__ -DMODULE -DMODVERSIONS
-include /usr/src/linux/include/linux/modversions.h -c
af_rfcomm.c
/home/haris/Diplwmtikh/affix-kernel-2.0.2/include/affix/hci.h:67:
warning: `sprintf_chk' defined but not used gcc -MD -O2 -Wall -
Wstrict-prototypes -fomit-frame-pointer -pipe -

```

```

I/home/haris/Diplwmtikh/affix-kernel-2.0.2/include -
I/usr/src/linux/include -include /home/haris/Diplwmtikh/affix-
kernel-2.0.2/include/affix.h -D__KERNEL__ -DMODULE -DMODVERSIONS
-include /usr/src/linux/include/linux/modversions.h -c bty.c
/home/haris/Diplwmtikh/affix-kernel-2.0.2/include/affix/hci.h:67:
warning: `sprintf_chk' defined but not used gcc -MD -O2 -Wall -
Wstrict-prototypes -fomit-frame-pointer -pipe -
I/home/haris/Diplwmtikh/affix-kernel-2.0.2/include -
I/usr/src/linux/include -include /home/haris/Diplwmtikh/affix-
kernel-2.0.2/include/affix.h -D__KERNEL__ -DMODULE -DMODVERSIONS
-include /usr/src/linux/include/linux/modversions.h -c sysctl-
rfcomm.c
/home/haris/Diplwmtikh/affix-kernel-2.0.2/include/affix/hci.h:67:
warning: `sprintf_chk' defined but not used sysctl-rfcomm.c:97:
warning: `affix_rfcomm_table' defined but not used
ld -r -o ../bin/affix_rfcomm.o rfcomm.o af_rfcomm.o bty.o sysctl-
rfcomm.o
chmod -x ../bin/affix_rfcomm.o
make[2]: Leaving directory `/home/haris/Diplwmtikh/affix-kernel-
2.0.2/rfcomm'
make[2]: Entering directory `/home/haris/Diplwmtikh/affix-kernel-
2.0.2/pan'
gcc -MD -O2 -Wall -Wstrict-prototypes -fomit-frame-pointer -pipe
-I/home/haris/Diplwmtikh/affix-kernel-2.0.2/include -
I/usr/src/linux/include -include /home/haris/Diplwmtikh/affix-
kernel-2.0.2/include/affix.h -D__KERNEL__ -DMODULE -DMODVERSIONS
-include /usr/src/linux/include/linux/modversions.h -c pan.c
pan.c: In function `pan_init':
pan.c:977: warning: implicit declaration of function `sscanf'
gcc -MD -O2 -Wall -Wstrict-prototypes -fomit-frame-pointer -pipe
-I/home/haris/Diplwmtikh/affix-kernel-2.0.2/include -
I/usr/src/linux/include -include /home/haris/Diplwmtikh/affix-
kernel-2.0.2/include/affix.h -D__KERNEL__ -DMODULE -DMODVERSIONS
-include /usr/src/linux/include/linux/modversions.h -c bnep.c
/home/haris/Diplwmtikh/affix-kernel-2.0.2/include/affix/hci.h:67:
warning: `sprintf_chk' defined but not used gcc -MD -O2 -Wall -
Wstrict-prototypes -fomit-frame-pointer -pipe -
I/home/haris/Diplwmtikh/affix-kernel-2.0.2/include -
I/usr/src/linux/include -include /home/haris/Diplwmtikh/affix-
kernel-2.0.2/include/affix.h -D__KERNEL__ -DMODULE -DMODVERSIONS
-include /usr/src/linux/include/linux/modversions.h -c sysctl-
pan.c
/home/haris/Diplwmtikh/affix-kernel-2.0.2/include/affix/hci.h:67:
warning: `sprintf_chk' defined but not used sysctl-pan.c:70:
warning: `affix_core_table' defined but not used
ld -r -o ../bin/affix_pan.o pan.o bnep.o sysctl-pan.o
chmod -x ../bin/affix_pan.o
make[2]: Leaving directory `/home/haris/Diplwmtikh/affix-kernel-
2.0.2/pan'
make[2]: Entering directory `/home/haris/Diplwmtikh/affix-kernel-
2.0.2/drivers/usb'
gcc -MD -O2 -Wall -Wstrict-prototypes -fomit-frame-pointer -pipe
-I/home/haris/Diplwmtikh/affix-kernel-2.0.2/include -
I/usr/src/linux/include -include /home/haris/Diplwmtikh/affix-
kernel-2.0.2/include/affix.h -D__KERNEL__ -DMODULE -DMODVERSIONS
-include /usr/src/linux/include/linux/modversions.h -c btusb.c
/home/haris/Diplwmtikh/affix-kernel-2.0.2/include/affix/hci.h:67:
warning: `sprintf_chk' defined but not used ld -r -o
../bin/affix_usb.o btusb.o
chmod -x ../bin/affix_usb.o

```

```

make[2]: Leaving directory `/home/haris/Diplwmtikh/affix-kernel-2.0.2/drivers/usb'
make[1]: Leaving directory `/home/haris/Diplwmtikh/affix-kernel-2.0.2'
[root@Haris affix-kernel-2.0.2]# make install
Checking kernel checksum ...
make[1]: Entering directory `/home/haris/Diplwmtikh/affix-kernel-2.0.2'
Compiling driver and control programs
make[2]: Entering directory `/home/haris/Diplwmtikh/affix-kernel-2.0.2/etc'
make[2]: Nothing to be done for `all'.
make[2]: Leaving directory `/home/haris/Diplwmtikh/affix-kernel-2.0.2/etc'
make[2]: Entering directory `/home/haris/Diplwmtikh/affix-kernel-2.0.2/include'
make[2]: Leaving directory `/home/haris/Diplwmtikh/affix-kernel-2.0.2/include'
make[2]: Entering directory `/home/haris/Diplwmtikh/affix-kernel-2.0.2/btcore'
make[2]: Nothing to be done for `all'.
make[2]: Leaving directory `/home/haris/Diplwmtikh/affix-kernel-2.0.2/btcore'
make[2]: Entering directory `/home/haris/Diplwmtikh/affix-kernel-2.0.2/rfcomm'
make[2]: Nothing to be done for `all'.
make[2]: Leaving directory `/home/haris/Diplwmtikh/affix-kernel-2.0.2/rfcomm'
make[2]: Entering directory `/home/haris/Diplwmtikh/affix-kernel-2.0.2/pan'
make[2]: Nothing to be done for `all'.
make[2]: Leaving directory `/home/haris/Diplwmtikh/affix-kernel-2.0.2/pan'
make[2]: Entering directory `/home/haris/Diplwmtikh/affix-kernel-2.0.2/drivers/usb'
make[2]: Nothing to be done for `all'.
make[2]: Leaving directory `/home/haris/Diplwmtikh/affix-kernel-2.0.2/drivers/usb'
make[1]: Leaving directory `/home/haris/Diplwmtikh/affix-kernel-2.0.2'
Removing old files from the system
# PCMCIA
rm -f /etc/pcmcia/affix_uart.conf
rm -f /etc/pcmcia/bt3c.conf /etc/pcmcia/bluecard.conf
# MODULES
rm -f /lib/modules/2.4.8-26mdk/net/affix*
rm -f /lib/modules/2.4.8-26mdk/pcmcia/affix*
rm -f /lib/modules/2.4.8-26mdk/kernel/net/affix_rfcomm.o
rm -f /lib/modules/2.4.8-26mdk/kernel/drivers/usb/affix_usb.o
rm -f /lib/modules/2.4.8-26mdk/kernel/drivers/bluetooth/affix*
rm -f /lib/modules/2.4.8-26mdk/kernel/net/bluetooth/affix*
#
rm -rf /usr/include/affix/types.h /usr/include/affix/debug.h
Installing driver and control programs
make[1]: Entering directory `/home/haris/Diplwmtikh/affix-kernel-2.0.2/etc'
installing devfs conf file
install -m 0755 -d /etc/devfs/conf.d
install -m 0644 devfs /etc/devfs/conf.d/affix
Installing scripts

```



```

install -m 0755 -d /etc/affix
install -m 0755 -d /etc/pcmcia
install -m 0644 pcmcia/*.conf /etc/pcmcia
make[1]: Leaving directory `/home/haris/Diplwmtikh/affix-kernel-2.0.2/etc'
make[1]: Entering directory `/home/haris/Diplwmtikh/affix-kernel-2.0.2/include'
install -m 0755 -d /usr/include/affix
for file in hci_types.h bluetooth.h hci_cmds.h ; do \
    diff -q $file /usr/include/affix/$file || \
        install -m 0644 $file /usr/include/affix; \
done
diff: /usr/include/affix/hci_types.h: No such file or directory
diff: /usr/include/affix/bluetooth.h: No such file or directory
diff: /usr/include/affix/hci_cmds.h: No such file or directory
make[1]: Leaving directory `/home/haris/Diplwmtikh/affix-kernel-2.0.2/include'
make[1]: Entering directory `/home/haris/Diplwmtikh/affix-kernel-2.0.2/btcore'
install -m 0755 -d /lib/modules/2.4.8-26mdk/kernel/net/affix
install -m 0644 ../bin/affix.o /lib/modules/2.4.8-26mdk/kernel/net/affix
make[1]: Leaving directory `/home/haris/Diplwmtikh/affix-kernel-2.0.2/btcore'
make[1]: Entering directory `/home/haris/Diplwmtikh/affix-kernel-2.0.2/rfcomm'
install -m 0755 -d /lib/modules/2.4.8-26mdk/kernel/net/affix
install -m 0644 ../bin/affix_rfcomm.o /lib/modules/2.4.8-26mdk/kernel/net/affix
make[1]: Leaving directory `/home/haris/Diplwmtikh/affix-kernel-2.0.2/rfcomm'
make[1]: Entering directory `/home/haris/Diplwmtikh/affix-kernel-2.0.2/pan'
install -m 0755 -d /lib/modules/2.4.8-26mdk/kernel/net/affix
install -m 0644 ../bin/affix_pan.o /lib/modules/2.4.8-26mdk/kernel/net/affix
make[1]: Leaving directory `/home/haris/Diplwmtikh/affix-kernel-2.0.2/pan'
make[1]: Entering directory `/home/haris/Diplwmtikh/affix-kernel-2.0.2/drivers/usb'
install -m 0755 -d /lib/modules/2.4.8-26mdk/kernel/drivers/affix
install -m 0644 ../../bin/affix_usb.o /lib/modules/2.4.8-26mdk/kernel/drivers/affix
make[1]: Leaving directory `/home/haris/Diplwmtikh/affix-kernel-2.0.2/drivers/usb'
depmod: *** Unresolved symbols in /lib/modules/2.4.8-26mdk/kernel/net/affix/affix_pan.o
Installation complete
[root@Haris affix-kernel-2.0.2]#

```

7.3.4 Εγκατάσταση του Affix

Αφού εγκαταστήσαμε επιτυχώς το affix-kernel μπορούμε να προχωρήσουμε στην εγκατάσταση του Affix. Βγαίνουμε από το φάκελο του affix-kernel και πάμε στο φάκελο που βρίσκεται το affix.

1. Το πρώτο βήμα, περιλαμβάνει το configure του αρχείου
Πληκτρολογούμε την εντολή:

➤ **./configure**

Μπορούμε να ελέγξουμε κάποιες παραμέτρους. Δίνουμε μερικές ενδεικτικά:

```
--enable-audio use audio (default is "yes")  
--enable-rfcomm use rfcomm (default is "yes")  
--enable-pan use pan (default is "yes")  
--enable-uart use uart (default is "yes")  
--enable-sdp use sdp (default is "yes")  
  
--enable-obex use obex (default is "yes")  
--enable-hfp use hfp (default is "no")  
--enable-debug use debug (default is "no")
```

Μπορείτε να τα δείτε πληκτρολογώντας την εντολή: `./configure --help`

2. Προχωράμε στο στάδιο του compiling
Αυτό γίνεται με την εντολή:

➤ **make all**

3. Και ολοκληρώνουμε τη διαδικασία εγκατάστασης με την εντολή:

✦ make install

με την οποία εγκαθίστανται οι απαραίτητες βιβλιοθήκες το πρόγραμμα διαχείρισης του affix, τα configuration files και άλλα.

Konsole haris@Haris.Alexis /home Konsole

```
[root@Haris haris]# cd /home/haris/Diplwmtikh/affix-2.0.2
[root@Haris affix-2.0.2]# ./configure
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking whether make sets $(MAKE)... yes
checking for working aclocal-1.4... missing
checking for working autoconf... found
checking for working automake-1.4... missing
checking for working autoheader... found
checking for working makeinfo... found
checking for gcc... gcc
checking for C compiler default output... a.out
checking whether the C compiler works... yes
checking whether we are cross compiling... no
checking for suffix of executables...
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ANSI C... none needed
checking build system type... i686-pc-linux-gnu
checking host system type... i686-pc-linux-gnu
checking for ld used by GCC... /usr/bin/ld
checking if the linker (/usr/bin/ld) is GNU ld... yes
checking for /usr/bin/ld option to reload object files... -r
checking for BSD-compatible nm... /usr/bin/nm -B
checking for a sed that does not truncate output... /bin/sed
checking whether ln -s works... yes
checking how to recognise dependent libraries... pass_all
checking command to parse /usr/bin/nm -B output... ok
checking how to run the C preprocessor... gcc -E
checking for egrep... grep -E
checking for ANSI C header files... yes
checking for sys/types.h... yes
checking for sys/stat.h... yes
checking for stdlib.h... yes
checking for string.h... yes
checking for memory.h... yes
checking for strings.h... yes
checking for inttypes.h... yes
checking for stdint.h... yes
checking for unistd.h... yes
checking dlfcn.h usability... yes
checking dlfcn.h presence... yes
checking for dlfcn.h... yes
checking for ranlib... ranlib
```

```

checking for strip... strip
checking for objdir... .libs
checking for gcc option to produce PIC... -fPIC
checking if gcc PIC flag -fPIC works... yes
checking if gcc static flag -static works... yes
checking if gcc supports -c -o file.o... yes
checking if gcc supports -c -o file.lo... yes
checking if gcc supports -fno-rtti -fno-exceptions... yes
checking whether the linker (/usr/bin/ld) supports shared
libraries... yes
checking how to hardcode library paths into programs... immediate
checking whether stripping libraries is possible... yes
checking dynamic linker characteristics... GNU/Linux ld.so
checking if libtool supports shared libraries... yes
checking whether to build shared libraries... yes
checking whether to build static libraries... yes
checking whether -lc should be explicitly linked in... no
creating libtool
checking for a BSD-compatible install... /usr/bin/install -c
checking for bison... bison -y
checking for flex... flex
checking for flex... (cached) flex
checking for yywrap in -lfl... yes
checking lex output file root... lex.yy
checking whether yytext is a pointer... yes
checking affix/bluetooth.h usability... yes
checking affix/bluetooth.h presence... yes
checking for affix/bluetooth.h... yes
checking whether to use audio... yes
checking whether to use rfcomm... yes
checking whether to use pan... yes
checking whether to use uart... yes
checking whether to use sdp... yes
checking whether to use obex... yes
checking whether to use debug... no
checking for openobex-config... /usr/bin/openobex-config
checking for openobex - version >= 1.0.0... yes (1.0.1)
configure: creating ./config.status
config.status: creating Makefile
config.status: creating lib/Makefile
config.status: creating sdp/Makefile
config.status: creating obex/Makefile
config.status: creating tools/Makefile
config.status: creating daemon/Makefile
config.status: creating control/Makefile
config.status: creating etc/Makefile
config.status: creating etc/btsrv.conf
config.status: creating etc/affix.conf
config.status: creating include/Makefile
config.status: creating include/config.h
config.status: linking ./include to include/affix
config.status: executing default-1 commands
[root@Haris affix-2.0.2]# make
Making all in lib
make[1]: Entering directory `/home/haris/Diplwmtikh/affix-
2.0.2/lib'
/bin/sh ../libtool --mode=compile gcc -DHAVE_CONFIG_H -I. -I. -
I../include -D_GNU_SOURCE -I../include -include
../include/config.h -O2 -Wall -Wstrict-prototypes -fomit-frame-
pointer -pipe -c btcore.c

```

```

mkdir .libs
gcc -DHAVE_CONFIG_H -I. -I. -I../include -D_GNU_SOURCE -
I../include -include ../include/config.h -O2 -Wall -Wstrict-
prototypes -fomit-frame-pointer -pipe -c btcore.c -fPIC -DPIC -o
.libs/btcore.lo
gcc -DHAVE_CONFIG_H -I. -I. -I../include -D_GNU_SOURCE -
I../include -include ../include/config.h -O2 -Wall -Wstrict-
prototypes -fomit-frame-pointer -pipe -c btcore.c -o btcore.o
>/dev/null 2>&1
mv -f .libs/btcore.lo btcore.lo
/bin/sh ../libtool --mode=compile gcc -DHAVE_CONFIG_H -I. -I. -
I../include -D_GNU_SOURCE -I../include -include
../include/config.h -O2 -Wall -Wstrict-prototypes -fomit-frame-
pointer -pipe -c btutils.c
rm -f .libs/btutils.lo
gcc -DHAVE_CONFIG_H -I. -I. -I../include -D_GNU_SOURCE -
I../include -include ../include/config.h -O2 -Wall -Wstrict-
prototypes -fomit-frame-pointer -pipe -c btutils.c -fPIC -DPIC -
o .libs/btutils.lo
gcc -DHAVE_CONFIG_H -I. -I. -I../include -D_GNU_SOURCE -
I../include -include ../include/config.h -O2 -Wall -Wstrict-
prototypes -fomit-frame-pointer -pipe -c btutils.c -o btutils.o
>/dev/null 2>&1
mv -f .libs/btutils.lo btutils.lo
/bin/sh ../libtool --mode=link gcc -O2 -Wall -Wstrict-
prototypes -fomit-frame-pointer -pipe -o libaffix.la -rpath
/usr/local/lib -version-info 2:2:0 btcore.lo btutils.lo
rm -fr .libs/libaffix.la .libs/libaffix.* .libs/libaffix.*
gcc -shared btcore.lo btutils.lo -Wl,-soname -Wl,libaffix.so.2
-o .libs/libaffix.so.2.0.2
(cd .libs && rm -f libaffix.so.2 && ln -s libaffix.so.2.0.2
libaffix.so.2)
(cd .libs && rm -f libaffix.so && ln -s libaffix.so.2.0.2
libaffix.so)
ar cru .libs/libaffix.a btcore.o btutils.o
ranlib .libs/libaffix.a
creating libaffix.la
(cd .libs && rm -f libaffix.la && ln -s ../libaffix.la
libaffix.la)
make[1]: Leaving directory `/home/haris/Diplwmtikh/affix-
2.0.2/lib'
Making all in sdp
make[1]: Entering directory `/home/haris/Diplwmtikh/affix-
2.0.2/sdp'
/bin/sh ../libtool --mode=compile gcc -DHAVE_CONFIG_H -I. -I. -
I../include -D_GNU_SOURCE -I../include -include
../include/config.h -DBTDEBUG_MODULE=BTDEBUG_MODULE_SDP -O2 -
Wall -Wstrict-prototypes -fomit-frame-pointer -pipe -c attr.c
mkdir .libs
gcc -DHAVE_CONFIG_H -I. -I. -I../include -D_GNU_SOURCE -
I../include -include ../include/config.h -
DBTDEBUG_MODULE=BTDEBUG_MODULE_SDP -O2 -Wall -Wstrict-prototypes
-fomit-frame-pointer -pipe -c attr.c -fPIC -DPIC -o
.libs/attr.lo
gcc -DHAVE_CONFIG_H -I. -I. -I../include -D_GNU_SOURCE -
I../include -include ../include/config.h -
DBTDEBUG_MODULE=BTDEBUG_MODULE_SDP -O2 -Wall -Wstrict-prototypes
-fomit-frame-pointer -pipe -c attr.c -o attr.o >/dev/null 2>&1
mv -f .libs/attr.lo attr.lo
/bin/sh ../libtool --mode=compile gcc -DHAVE_CONFIG_H -I. -I. -

```

```

I../include -D_GNU_SOURCE -I../include -include
../include/config.h -DBTDEBUG_MODULE=BTDEBUG_MODULE_SDP -O2 -
Wall -Wstrict-prototypes -fomit-frame-pointer -pipe -c uuid.c
rm -f .libs/uuid.lo
gcc -DHAVE_CONFIG_H -I. -I. -I../include -D_GNU_SOURCE -
I../include -include ../include/config.h -
DBTDEBUG_MODULE=BTDEBUG_MODULE_SDP -O2 -Wall -Wstrict-prototypes
-fomit-frame-pointer -pipe -c uuid.c -fPIC -DPIC -o
.libs/uuid.lo
gcc -DHAVE_CONFIG_H -I. -I. -I../include -D_GNU_SOURCE -
I../include -include ../include/config.h -
DBTDEBUG_MODULE=BTDEBUG_MODULE_SDP -O2 -Wall -Wstrict-prototypes
-fomit-frame-pointer -pipe -c uuid.c -o uuid.o >/dev/null 2>&1
mv -f .libs/uuid.lo uuid.lo
/bin/sh ../libtool --mode=compile gcc -DHAVE_CONFIG_H -I. -I. -
I../include -D_GNU_SOURCE -I../include -include
../include/config.h -DBTDEBUG_MODULE=BTDEBUG_MODULE_SDP -O2 -
Wall -Wstrict-prototypes -fomit-frame-pointer -pipe -c des.c
rm -f .libs/des.lo
gcc -DHAVE_CONFIG_H -I. -I. -I../include -D_GNU_SOURCE -
I../include -include ../include/config.h -
DBTDEBUG_MODULE=BTDEBUG_MODULE_SDP -O2 -Wall -Wstrict-prototypes
-fomit-frame-pointer -pipe -c des.c -fPIC -DPIC -o .libs/des.lo
gcc -DHAVE_CONFIG_H -I. -I. -I../include -D_GNU_SOURCE -
I../include -include ../include/config.h -
DBTDEBUG_MODULE=BTDEBUG_MODULE_SDP -O2 -Wall -Wstrict-prototypes
-fomit-frame-pointer -pipe -c des.c -o des.o >/dev/null 2>&1
mv -f .libs/des.lo des.lo
/bin/sh ../libtool --mode=compile gcc -DHAVE_CONFIG_H -I. -I. -
I../include -D_GNU_SOURCE -I../include -include
../include/config.h -DBTDEBUG_MODULE=BTDEBUG_MODULE_SDP -O2 -
Wall -Wstrict-prototypes -fomit-frame-pointer -pipe -c utils.c
rm -f .libs/utils.lo
gcc -DHAVE_CONFIG_H -I. -I. -I../include -D_GNU_SOURCE -
I../include -include ../include/config.h -
DBTDEBUG_MODULE=BTDEBUG_MODULE_SDP -O2 -Wall -Wstrict-prototypes
-fomit-frame-pointer -pipe -c utils.c -fPIC -DPIC -o
.libs/utils.lo
gcc -DHAVE_CONFIG_H -I. -I. -I../include -D_GNU_SOURCE -
I../include -include ../include/config.h -
DBTDEBUG_MODULE=BTDEBUG_MODULE_SDP -O2 -Wall -Wstrict-prototypes
-fomit-frame-pointer -pipe -c utils.c -o utils.o >/dev/null 2>&1
mv -f .libs/utils.lo utils.lo
/bin/sh ../libtool --mode=compile gcc -DHAVE_CONFIG_H -I. -I. -
I../include -D_GNU_SOURCE -I../include -include
../include/config.h -DBTDEBUG_MODULE=BTDEBUG_MODULE_SDP -O2 -
Wall -Wstrict-prototypes -fomit-frame-pointer -pipe -c sdpclt.c
rm -f .libs/sdpclt.lo
gcc -DHAVE_CONFIG_H -I. -I. -I../include -D_GNU_SOURCE -
I../include -include ../include/config.h -
DBTDEBUG_MODULE=BTDEBUG_MODULE_SDP -O2 -Wall -Wstrict-prototypes
-fomit-frame-pointer -pipe -c sdpclt.c -fPIC -DPIC -o
.libs/sdpclt.lo
gcc -DHAVE_CONFIG_H -I. -I. -I../include -D_GNU_SOURCE -
I../include -include ../include/config.h -
DBTDEBUG_MODULE=BTDEBUG_MODULE_SDP -O2 -Wall -Wstrict-prototypes
-fomit-frame-pointer -pipe -c sdpclt.c -o sdpclt.o >/dev/null
2>&1
mv -f .libs/sdpclt.lo sdpclt.lo
/bin/sh ../libtool --mode=compile gcc -DHAVE_CONFIG_H -I. -I. -

```

```

I../include -D_GNU_SOURCE -I../include -include
../include/config.h -DBTDEBUG_MODULE=BTDEBUG_MODULE_SDP -O2 -
Wall -Wstrict-prototypes -fomit-frame-pointer -pipe -c cstate.c
rm -f .libs/cstate.lo
gcc -DHAVE_CONFIG_H -I. -I. -I../include -D_GNU_SOURCE -
I../include -include ../include/config.h -
DBTDEBUG_MODULE=BTDEBUG_MODULE_SDP -O2 -Wall -Wstrict-prototypes
-fomit-frame-pointer -pipe -c cstate.c -fPIC -DPIC -o
.libs/cstate.lo
gcc -DHAVE_CONFIG_H -I. -I. -I../include -D_GNU_SOURCE -
I../include -include ../include/config.h -
DBTDEBUG_MODULE=BTDEBUG_MODULE_SDP -O2 -Wall -Wstrict-prototypes
-fomit-frame-pointer -pipe -c cstate.c -o cstate.o >/dev/null
2>&1
mv -f .libs/cstate.lo cstate.lo
/bin/sh ../libtool --mode=link gcc -O2 -Wall -Wstrict-
prototypes -fomit-frame-pointer -pipe -o libaffix_sdp.la -rpath
/usr/local/lib -version-info 2:2:0 attr.lo uuid.lo des.lo
utils.lo sdpc.lt.lo cstate.lo ../lib/libaffix.la
rm -fr .libs/libaffix_sdp.la .libs/libaffix_sdp.*
.libs/libaffix_sdp.*
gcc -shared attr.lo uuid.lo des.lo utils.lo sdpc.lt.lo cstate.lo
-Wl,--rpath -Wl,/home/haris/Diplwmtikh/affix-2.0.2/lib/.libs -
Wl,--rpath -Wl,/usr/local/lib ../lib/.libs/libaffix.so -Wl,-
soname -Wl,libaffix_sdp.so.2 -o .libs/libaffix_sdp.so.2.0.2
(cd .libs && rm -f libaffix_sdp.so.2 && ln -s
libaffix_sdp.so.2.0.2 libaffix_sdp.so.2)
(cd .libs && rm -f libaffix_sdp.so && ln -s libaffix_sdp.so.2.0.2
libaffix_sdp.so)
ar cru .libs/libaffix_sdp.a attr.o uuid.o des.o utils.o sdpc.lt.o
cstate.o
ranlib .libs/libaffix_sdp.a
creating libaffix_sdp.la
(cd .libs && rm -f libaffix_sdp.la && ln -s ../libaffix_sdp.la
libaffix_sdp.la)
/bin/sh ../libtool --mode=compile gcc -DHAVE_CONFIG_H -I. -I. -
I../include -D_GNU_SOURCE -I../include -include
../include/config.h -DBTDEBUG_MODULE=BTDEBUG_MODULE_SDP -O2 -
Wall -Wstrict-prototypes -fomit-frame-pointer -pipe -c sdpsrv.c
rm -f .libs/sdpsrv.lo
gcc -DHAVE_CONFIG_H -I. -I. -I../include -D_GNU_SOURCE -
I../include -include ../include/config.h -
DBTDEBUG_MODULE=BTDEBUG_MODULE_SDP -O2 -Wall -Wstrict-prototypes
-fomit-frame-pointer -pipe -c sdpsrv.c -fPIC -DPIC -o
.libs/sdpsrv.lo
gcc -DHAVE_CONFIG_H -I. -I. -I../include -D_GNU_SOURCE -
I../include -include ../include/config.h -
DBTDEBUG_MODULE=BTDEBUG_MODULE_SDP -O2 -Wall -Wstrict-prototypes
-fomit-frame-pointer -pipe -c sdpsrv.c -o sdpsrv.o >/dev/null
2>&1
mv -f .libs/sdpsrv.lo sdpsrv.lo
/bin/sh ../libtool --mode=link gcc -O2 -Wall -Wstrict-
prototypes -fomit-frame-pointer -pipe -o libaffix_sdpsvc.la -
rpath /usr/local/lib -version-info 2:2:0 sdpsrv.lo
libaffix_sdp.la ../lib/libaffix.la
rm -fr .libs/libaffix_sdpsvc.la .libs/libaffix_sdpsvc.*
.libs/libaffix_sdpsvc.*
gcc -shared sdpsrv.lo -Wl,--rpath -
Wl,/home/haris/Diplwmtikh/affix-2.0.2/sdp/.libs -Wl,--rpath -
Wl,/home/haris/Diplwmtikh/affix-2.0.2/lib/.libs -Wl,--rpath -

```

```

Wl,/usr/local/lib -L/home/haris/Diplwmtikh/affix-2.0.2/lib/.libs
./libs/libaffix_sdp.so ../lib/.libs/libaffix.so -Wl,-soname -
Wl,libaffix_sdpsvc.so.2 -o .libs/libaffix_sdpsvc.so.2.0.2
(cd .libs && rm -f libaffix_sdpsvc.so.2 && ln -s
libaffix_sdpsvc.so.2.0.2 libaffix_sdpsvc.so.2)
(cd .libs && rm -f libaffix_sdpsvc.so && ln -s
libaffix_sdpsvc.so.2.0.2 libaffix_sdpsvc.so)
ar cru .libs/libaffix_sdpsvc.a sdpsrv.o
ranlib .libs/libaffix_sdpsvc.a
creating libaffix_sdpsvc.la
(cd .libs && rm -f libaffix_sdpsvc.la && ln -s
../libaffix_sdpsvc.la libaffix_sdpsvc.la)
gcc -DHAVE_CONFIG_H -I. -I. -I../include -D_GNU_SOURCE -
I../include -include ../include/config.h -
DBTDEBUG_MODULE=BTDEBUG_MODULE_SDP -O2 -Wall -Wstrict-prototypes
-fomit-frame-pointer -pipe -c sdpserver.c
gcc -DHAVE_CONFIG_H -I. -I. -I../include -D_GNU_SOURCE -
I../include -include ../include/config.h -
DBTDEBUG_MODULE=BTDEBUG_MODULE_SDP -O2 -Wall -Wstrict-prototypes
-fomit-frame-pointer -pipe -c servicedb.c
gcc -DHAVE_CONFIG_H -I. -I. -I../include -D_GNU_SOURCE -
I../include -include ../include/config.h -
DBTDEBUG_MODULE=BTDEBUG_MODULE_SDP -O2 -Wall -Wstrict-prototypes
-fomit-frame-pointer -pipe -c request.c
/bin/sh ../libtool --mode=link gcc -O2 -Wall -Wstrict-
prototypes -fomit-frame-pointer -pipe -o btsdp sdpserver.o
servicedb.o request.o libaffix_sdp.la libaffix_sdpsvc.la
../lib/libaffix.la
gcc -O2 -Wall -Wstrict-prototypes -fomit-frame-pointer -pipe -o
.libs/btsdp sdpserver.o servicedb.o request.o
../libs/libaffix_sdp.so ../libs/libaffix_sdpsvc.so
/home/haris/Diplwmtikh/affix-2.0.2/sdp/.libs/libaffix_sdp.so
/home/haris/Diplwmtikh/affix-2.0.2/lib/.libs/libaffix.so
../lib/.libs/libaffix.so -Wl,--rpath -Wl,/usr/local/lib
creating btsdp
make[1]: Leaving directory `/home/haris/Diplwmtikh/affix-
2.0.2/sdp'
Making all in obex
make[1]: Entering directory `/home/haris/Diplwmtikh/affix-
2.0.2/obex'
/bin/sh ../libtool --mode=compile gcc -DHAVE_CONFIG_H -I. -I. -
I../include -D_GNU_SOURCE -I../include -include
../include/config.h -I/usr/include -
DBTDEBUG_MODULE=BTDEBUG_MODULE_OBEX -O2 -Wall -Wstrict-
prototypes -fomit-frame-pointer -pipe -c obex_client.c
mkdir .libs
gcc -DHAVE_CONFIG_H -I. -I. -I../include -D_GNU_SOURCE -
I../include -include ../include/config.h -I/usr/include -
DBTDEBUG_MODULE=BTDEBUG_MODULE_OBEX -O2 -Wall -Wstrict-prototypes
-fomit-frame-pointer -pipe -c obex_client.c -fPIC -DPIC -o
.libs/obex_client.lo
gcc -DHAVE_CONFIG_H -I. -I. -I../include -D_GNU_SOURCE -
I../include -include ../include/config.h -I/usr/include -
DBTDEBUG_MODULE=BTDEBUG_MODULE_OBEX -O2 -Wall -Wstrict-prototypes
-fomit-frame-pointer -pipe -c obex_client.c -o obex_client.o
>/dev/null 2>&1
mv -f .libs/obex_client.lo obex_client.lo
/bin/sh ../libtool --mode=compile gcc -DHAVE_CONFIG_H -I. -I. -
I../include -D_GNU_SOURCE -I../include -include
../include/config.h -I/usr/include -

```



```

DBTDEBUG_MODULE=BTDEBUG_MODULE_OBEX -O2 -Wall -Wstrict-
prototypes -fomit-frame-pointer -pipe -c obex_server.c
rm -f .libs/obex_server.lo
gcc -DHAVE_CONFIG_H -I. -I. -I../include -D_GNU_SOURCE -
I../include -include ../include/config.h -I/usr/include -
DBTDEBUG_MODULE=BTDEBUG_MODULE_OBEX -O2 -Wall -Wstrict-prototypes
-fomit-frame-pointer -pipe -c obex_server.c -fPIC -DPIC -o
.libs/obex_server.lo
gcc -DHAVE_CONFIG_H -I. -I. -I../include -D_GNU_SOURCE -
I../include -include ../include/config.h -I/usr/include -
DBTDEBUG_MODULE=BTDEBUG_MODULE_OBEX -O2 -Wall -Wstrict-prototypes
-fomit-frame-pointer -pipe -c obex_server.c -o obex_server.o
>/dev/null 2>&1
mv -f .libs/obex_server.lo obex_server.lo
/bin/sh ../libtool --mode=link gcc -O2 -Wall -Wstrict-
prototypes -fomit-frame-pointer -pipe -o libaffix_obex.la -rpath
/usr/local/lib -version-info 2:2:0 obex_client.lo obex_server.lo
../sdp/libaffix_sdp.la ../sdp/libaffix_sdpsvc.la
../lib/libaffix.la -lopenobex
rm -fr .libs/libaffix_obex.la .libs/libaffix_obex.*
.libs/libaffix_obex.*
gcc -shared obex_client.lo obex_server.lo -Wl,--rpath -
Wl,/home/haris/Diplwmtikh/affix-2.0.2/sdp/.libs -Wl,--rpath -
Wl,/home/haris/Diplwmtikh/affix-2.0.2/lib/.libs -Wl,--rpath -
Wl,/usr/local/lib -L/home/haris/Diplwmtikh/affix-2.0.2/sdp/.libs
-L/home/haris/Diplwmtikh/affix-2.0.2/lib/.libs
../sdp/.libs/libaffix_sdp.so ../sdp/.libs/libaffix_sdpsvc.so
../lib/.libs/libaffix.so /usr/lib/libopenobex.so -Wl,-soname
-Wl,libaffix_obex.so.2 -o .libs/libaffix_obex.so.2.0.2
(cd .libs && rm -f libaffix_obex.so.2 && ln -s
libaffix_obex.so.2.0.2 libaffix_obex.so.2)
(cd .libs && rm -f libaffix_obex.so && ln -s
libaffix_obex.so.2.0.2 libaffix_obex.so)
ar cru .libs/libaffix_obex.a obex_client.o obex_server.o
ranlib .libs/libaffix_obex.a
creating libaffix_obex.la
(cd .libs && rm -f libaffix_obex.la && ln -s ../libaffix_obex.la
libaffix_obex.la)
gcc -DHAVE_CONFIG_H -I. -I. -I../include -D_GNU_SOURCE -
I../include -include ../include/config.h -I/usr/include -
DBTDEBUG_MODULE=BTDEBUG_MODULE_OBEX -O2 -Wall -Wstrict-
prototypes -fomit-frame-pointer -pipe -c btobex.c/bin/sh
../libtool --mode=link gcc -O2 -Wall -Wstrict-prototypes -
fomit-frame-pointer -pipe -o btobex btobex.o libaffix_obex.la
../lib/libaffix.la -lopenobex
gcc -O2 -Wall -Wstrict-prototypes -fomit-frame-pointer -pipe -o
.libs/btobex btobex.o ../libs/libaffix_obex.so
/home/haris/Diplwmtikh/affix-2.0.2/sdp/.libs/libaffix_sdpsvc.so
/home/haris/Diplwmtikh/affix-2.0.2/sdp/.libs/libaffix_sdp.so
/home/haris/Diplwmtikh/affix-2.0.2/lib/.libs/libaffix.so
../lib/.libs/libaffix.so /usr/lib/libopenobex.so -Wl,--rpath -
Wl,/usr/local/lib
creating btobex
make[1]: Leaving directory `/home/haris/Diplwmtikh/affix-
2.0.2/obex'
Making all in tools
make[1]: Entering directory `/home/haris/Diplwmtikh/affix-
2.0.2/tools'
gcc -DHAVE_CONFIG_H -I. -I. -I../include -D_GNU_SOURCE -
I../include -include ../include/config.h -I/usr/include -

```

```

DBTDEBUG_MODULE=BTDEBUG_MODULE_TOOLS -O2 -Wall -Wstrict-
prototypes -fomit-frame-pointer -pipe -c btmodem.c
/bin/sh ../libtool --mode=link gcc -O2 -Wall -Wstrict-
prototypes -fomit-frame-pointer -pipe -o btmodem btmodem.o
../lib/libaffix.la
mkdir .libs
gcc -O2 -Wall -Wstrict-prototypes -fomit-frame-pointer -pipe -o
.libs/btmodem btmodem.o ../lib/.libs/libaffix.so -Wl,--rpath -
Wl,/usr/local/lib
creating btmodem
make[1]: Leaving directory `/home/haris/Diplwmtikh/affix-
2.0.2/tools'
Making all in daemon
make[1]: Entering directory `/home/haris/Diplwmtikh/affix-
2.0.2/daemon'
gcc -DHAVE_CONFIG_H -I. -I. -I../include -D_GNU_SOURCE -
I../include -include ../include/config.h -I/usr/include -
DBTDEBUG_MODULE=BTDEBUG_MODULE_CONTROL -O2 -Wall -Wstrict-
prototypes -fomit-frame-pointer -pipe -c btsrv.c
gcc -DHAVE_CONFIG_H -I. -I. -I../include -D_GNU_SOURCE -
I../include -include ../include/config.h -I/usr/include -
DBTDEBUG_MODULE=BTDEBUG_MODULE_CONTROL -O2 -Wall -Wstrict-
prototypes -fomit-frame-pointer -pipe -c btsrv-parser.c
gcc -DHAVE_CONFIG_H -I. -I. -I../include -D_GNU_SOURCE -
I../include -include ../include/config.h -I/usr/include -
DBTDEBUG_MODULE=BTDEBUG_MODULE_CONTROL -O2 -Wall -Wstrict-
prototypes -fomit-frame-pointer -pipe -c btsrv-lexer.c
lex.yy.c: In function `yylex':
lex.yy.c:778: warning: label `find_rule' defined but not used
btsrv-lexer.l: At top level:
lex.yy.c:1841: warning: `yy_flex_realloc' defined but not used
lex.yy.c:1320: warning: `yyunput' defined but not used
gcc -DHAVE_CONFIG_H -I. -I. -I../include -D_GNU_SOURCE -
I../include -include ../include/config.h -I/usr/include -
DBTDEBUG_MODULE=BTDEBUG_MODULE_CONTROL -O2 -Wall -Wstrict-
prototypes -fomit-frame-pointer -pipe -c btsrv-sdp.c
/bin/sh ../libtool --mode=link gcc -O2 -Wall -Wstrict-
prototypes -fomit-frame-pointer -pipe -o btsrv btsrv.o btsrv-
parser.o btsrv-lexer.o ../sdp/libaffix_sdp.la
../sdp/libaffix_sdpsvc.la btsrv-sdp.o ../lib/libaffix.la
mkdir .libs
gcc -O2 -Wall -Wstrict-prototypes -fomit-frame-pointer -pipe -o
.libs/btsrv btsrv.o btsrv-parser.o btsrv-lexer.o btsrv-sdp.o
../sdp/.libs/libaffix_sdp.so ../sdp/.libs/libaffix_sdpsvc.so
/home/haris/Diplwmtikh/affix-2.0.2/sdp/.libs/libaffix_sdp.so
/home/haris/Diplwmtikh/affix-2.0.2/lib/.libs/libaffix.so
../lib/.libs/libaffix.so -Wl,--rpath -Wl,/usr/local/lib
creating btsrv
make[1]: Leaving directory `/home/haris/Diplwmtikh/affix-
2.0.2/daemon'
Making all in control
make[1]: Entering directory `/home/haris/Diplwmtikh/affix-
2.0.2/control'
gcc -DHAVE_CONFIG_H -I. -I. -I../include -D_GNU_SOURCE -
I../include -include ../include/config.h -I/usr/include -
DBTDEBUG_MODULE=BTDEBUG_MODULE_CONTROL -O2 -Wall -Wstrict-
prototypes -fomit-frame-pointer -pipe -c btctl.c
gcc -DHAVE_CONFIG_H -I. -I. -I../include -D_GNU_SOURCE -
I../include -include ../include/config.h -I/usr/include -
DBTDEBUG_MODULE=BTDEBUG_MODULE_CONTROL -O2 -Wall -Wstrict-

```

```

prototypes -fomit-frame-pointer -pipe -c btctl-dev.c
gcc -DHAVE_CONFIG_H -I. -I. -I../include -D_GNU_SOURCE -
I../include -include ../include/config.h -I/usr/include -
DBTDEBUG_MODULE=BTDEBUG_MODULE_CONTROL -O2 -Wall -Wstrict-
prototypes -fomit-frame-pointer -pipe -c btctl-audio.c
gcc -DHAVE_CONFIG_H -I. -I. -I../include -D_GNU_SOURCE -
I../include -include ../include/config.h -I/usr/include -
DBTDEBUG_MODULE=BTDEBUG_MODULE_CONTROL -O2 -Wall -Wstrict-
prototypes -fomit-frame-pointer -pipe -c btctl-pan.c
gcc -DHAVE_CONFIG_H -I. -I. -I../include -D_GNU_SOURCE -
I../include -include ../include/config.h -I/usr/include -
DBTDEBUG_MODULE=BTDEBUG_MODULE_CONTROL -O2 -Wall -Wstrict-
prototypes -fomit-frame-pointer -pipe -c btctl-sdp.c
gcc -DHAVE_CONFIG_H -I. -I. -I../include -D_GNU_SOURCE -
I../include -include ../include/config.h -I/usr/include -
DBTDEBUG_MODULE=BTDEBUG_MODULE_CONTROL -O2 -Wall -Wstrict-
prototypes -fomit-frame-pointer -pipe -c btctl-obex.c
/bin/sh ../libtool --mode=link gcc -O2 -Wall -Wstrict-
prototypes -fomit-frame-pointer -pipe -o btctl btctl.o btctl-
dev.o btctl-audio.o ../obex/libaffix_obex.la btctl-pan.o
../sdp/libaffix_sdp.la ../sdp/libaffix_sdpsvc.la btctl-sdp.o
../obex/libaffix_obex.la btctl-obex.o ../lib/libaffix.la
mkdir .libs
gcc -O2 -Wall -Wstrict-prototypes -fomit-frame-pointer -pipe -o
.libs/btctl btctl.o btctl-dev.o btctl-audio.o btctl-pan.o btctl-
sdp.o btctl-obex.o ../sdp/.libs/libaffix_sdp.so
../sdp/.libs/libaffix_sdpsvc.so ../obex/.libs/libaffix_obex.so
/home/haris/Diplwmtikh/affix-2.0.2/sdp/.libs/libaffix_sdpsvc.so
/home/haris/Diplwmtikh/affix-2.0.2/sdp/.libs/libaffix_sdp.so
/home/haris/Diplwmtikh/affix-2.0.2/lib/.libs/libaffix.so
/usr/lib/libopenobex.so ../lib/.libs/libaffix.so -Wl,--rpath -
Wl,/usr/local/lib
creating btctl
make[1]: Leaving directory `/home/haris/Diplwmtikh/affix-
2.0.2/control'
Making all in etc
make[1]: Entering directory `/home/haris/Diplwmtikh/affix-
2.0.2/etc'
make[1]: Nothing to be done for `all'.
make[1]: Leaving directory `/home/haris/Diplwmtikh/affix-
2.0.2/etc'
Making all in include
make[1]: Entering directory `/home/haris/Diplwmtikh/affix-
2.0.2/include'
make[1]: Leaving directory `/home/haris/Diplwmtikh/affix-
2.0.2/include'
make[1]: Entering directory `/home/haris/Diplwmtikh/affix-2.0.2'
make[1]: Nothing to be done for `all-am'.
make[1]: Leaving directory `/home/haris/Diplwmtikh/affix-2.0.2'
[root@Haris affix-2.0.2]# make install
Making install in lib
make[1]: Entering directory `/home/haris/Diplwmtikh/affix-
2.0.2/lib'
make[2]: Entering directory `/home/haris/Diplwmtikh/affix-
2.0.2/lib'
/bin/sh ../mkinstalldirs /usr/local/lib
/bin/sh ../libtool --mode=install /usr/bin/install -c
libaffix.la /usr/local/lib/libaffix.la
/usr/bin/install -c .libs/libaffix.so.2.0.2
/usr/local/lib/libaffix.so.2.0.2

```

```

(cd /usr/local/lib && rm -f libaffix.so.2 && ln -s
libaffix.so.2.0.2 libaffix.so.2)
(cd /usr/local/lib && rm -f libaffix.so && ln -s
libaffix.so.2.0.2 libaffix.so)
/usr/bin/install -c .libs/libaffix.lai /usr/local/lib/libaffix.la
/usr/bin/install -c .libs/libaffix.a /usr/local/lib/libaffix.a
ranlib /usr/local/lib/libaffix.a
chmod 644 /usr/local/lib/libaffix.a
PATH="$PATH:/sbin" ldconfig -n /usr/local/lib
-----
-----
Libraries have been installed in:
  /usr/local/lib

If you ever happen to want to link against installed libraries
in a given directory, LIBDIR, you must either use libtool, and
specify the full pathname of the library, or use the '-LLIBDIR'
flag during linking and do at least one of the following:
  - add LIBDIR to the 'LD_LIBRARY_PATH' environment variable
    during execution
  - add LIBDIR to the 'LD_RUN_PATH' environment variable
    during linking
  - use the '-Wl,--rpath -Wl,LIBDIR' linker flag
  - have your system administrator add LIBDIR to
    '/etc/ld.so.conf'

See any operating system documentation about shared libraries for
more information, such as the ld(1) and ld.so(8) manual pages.
-----
-----
make[2]: Nothing to be done for `install-data-am'.
make[2]: Leaving directory `/home/haris/Diplwmtikh/affix-
2.0.2/lib'
make[1]: Leaving directory `/home/haris/Diplwmtikh/affix-
2.0.2/lib'
Making install in sdp
make[1]: Entering directory `/home/haris/Diplwmtikh/affix-
2.0.2/sdp'
make[2]: Entering directory `/home/haris/Diplwmtikh/affix-
2.0.2/sdp'
/bin/sh ../mkinstalldirs /usr/local/lib
/bin/sh ../libtool  --mode=install /usr/bin/install -c
libaffix_sdp.la /usr/local/lib/libaffix_sdp.la
libtool: install: warning: relinking `libaffix_sdp.la'
(cd /home/haris/Diplwmtikh/affix-2.0.2/sdp; /bin/sh ../libtool --
mode=relink gcc -O2 -Wall -Wstrict-prototypes -fomit-frame-
pointer -pipe -o libaffix_sdp.la -rpath /usr/local/lib -version-
info 2:2:0 attr.lo uuid.lo des.lo utils.lo sdpclt.lo cstate.lo
../lib/libaffix.la )
gcc -shared attr.lo uuid.lo des.lo utils.lo sdpclt.lo cstate.lo
-Wl,--rpath -Wl,/usr/local/lib -L/usr/local/lib -laffix -Wl,-
soname -Wl,libaffix_sdp.so.2 -o .libs/libaffix_sdp.so.2.0.2
/usr/bin/install -c .libs/libaffix_sdp.so.2.0.2T
/usr/local/lib/libaffix_sdp.so.2.0.2
(cd /usr/local/lib && rm -f libaffix_sdp.so.2 && ln -s
libaffix_sdp.so.2.0.2 libaffix_sdp.so.2)
(cd /usr/local/lib && rm -f libaffix_sdp.so && ln -s
libaffix_sdp.so.2.0.2 libaffix_sdp.so)
/usr/bin/install -c .libs/libaffix_sdp.lai
/usr/local/lib/libaffix_sdp.la

```

```

/usr/bin/install -c .libs/libaffix_sdp.a
/usr/local/lib/libaffix_sdp.a
ranlib /usr/local/lib/libaffix_sdp.a
chmod 644 /usr/local/lib/libaffix_sdp.a
PATH="$PATH:/sbin" ldconfig -n /usr/local/lib
-----
-----
Libraries have been installed in:
    /usr/local/lib

If you ever happen to want to link against installed libraries
in a given directory, LIBDIR, you must either use libtool, and
specify the full pathname of the library, or use the '-LLIBDIR'
flag during linking and do at least one of the following:
    - add LIBDIR to the 'LD_LIBRARY_PATH' environment variable
      during execution
    - add LIBDIR to the 'LD_RUN_PATH' environment variable
      during linking
    - use the '-Wl,--rpath -Wl,LIBDIR' linker flag
    - have your system administrator add LIBDIR to
      '/etc/ld.so.conf'

See any operating system documentation about shared libraries for
more information, such as the ld(1) and ld.so(8) manual pages.
-----
-----
/bin/sh ../libtool  --mode=install /usr/bin/install -c
libaffix_sdpsvc.la /usr/local/lib/libaffix_sdpsvc.la
libtool: install: warning: relinking `libaffix_sdpsvc.la'
(cd /home/haris/Diplwmtikh/affix-2.0.2/sdp; /bin/sh ../libtool --
mode=relink gcc -O2 -Wall -Wstrict-prototypes -fomit-frame-
pointer -pipe -o libaffix_sdpsvc.la -rpath /usr/local/lib -
version-info 2:2:0 sdpsrv.lo libaffix_sdp.la ../lib/libaffix.la )
gcc -shared sdpsrv.lo -Wl,--rpath -Wl,/usr/local/lib -
L/home/haris/Diplwmtikh/affix-2.0.2/lib/.libs -L/usr/local/lib -
laffix_sdp -laffix -Wl,-soname -Wl,libaffix_sdpsvc.so.2 -o
.libs/libaffix_sdpsvc.so.2.0.2
/usr/bin/install -c .libs/libaffix_sdpsvc.so.2.0.2T
/usr/local/lib/libaffix_sdpsvc.so.2.0.2
(cd /usr/local/lib && rm -f libaffix_sdpsvc.so.2 && ln -s
libaffix_sdpsvc.so.2.0.2 libaffix_sdpsvc.so.2)
(cd /usr/local/lib && rm -f libaffix_sdpsvc.so && ln -s
libaffix_sdpsvc.so.2.0.2 libaffix_sdpsvc.so)
/usr/bin/install -c .libs/libaffix_sdpsvc.lai
/usr/local/lib/libaffix_sdpsvc.la
/usr/bin/install -c .libs/libaffix_sdpsvc.a
/usr/local/lib/libaffix_sdpsvc.a
ranlib /usr/local/lib/libaffix_sdpsvc.a
chmod 644 /usr/local/lib/libaffix_sdpsvc.a
PATH="$PATH:/sbin" ldconfig -n /usr/local/lib
-----
-----
Libraries have been installed in:
    /usr/local/lib

If you ever happen to want to link against installed libraries
in a given directory, LIBDIR, you must either use libtool, and
specify the full pathname of the library, or use the '-LLIBDIR'
flag during linking and do at least one of the following:
    - add LIBDIR to the 'LD_LIBRARY_PATH' environment variable

```

```

    during execution
- add LIBDIR to the `LD_RUN_PATH' environment variable
  during linking
- use the `-Wl,--rpath -Wl,LIBDIR' linker flag
- have your system administrator add LIBDIR to
`/etc/ld.so.conf'

```

See any operating system documentation about shared libraries for more information, such as the ld(1) and ld.so(8) manual pages.

```

-----
-----

```

```

/bin/sh ../mkinstalldirs /usr/local/bin
/bin/sh ../libtool  --mode=install /usr/bin/install -c  btsdp
/usr/local/bin/btsdp
/usr/bin/install -c .libs/btsdp /usr/local/bin/btsdp
make[2]: Nothing to be done for `install-data-am'.
make[2]: Leaving directory `/home/haris/Diplwmtikh/affix-
2.0.2/sdp'
make[1]: Leaving directory `/home/haris/Diplwmtikh/affix-
2.0.2/sdp'
Making install in obex
make[1]: Entering directory `/home/haris/Diplwmtikh/affix-
2.0.2/obex'
make[2]: Entering directory `/home/haris/Diplwmtikh/affix-
2.0.2/obex'
/bin/sh ../mkinstalldirs /usr/local/lib
/bin/sh ../libtool  --mode=install /usr/bin/install -c
libaffix_obex.la /usr/local/lib/libaffix_obex.la
libtool: install: warning: relinking `libaffix_obex.la'
(cd /home/haris/Diplwmtikh/affix-2.0.2/obex; /bin/sh ../libtool -
-mode=relink gcc -O2 -Wall -Wstrict-prototypes -fomit-frame-
pointer -pipe -o libaffix_obex.la -rpath /usr/local/lib -version-
info 2:2:0 obex_client.lo obex_server.lo ../sdp/libaffix_sdp.la
../sdp/libaffix_sdpsvc.la ../lib/libaffix.la -lopenobex )
gcc -shared obex_client.lo obex_server.lo -Wl,--rpath -
Wl,/usr/local/lib -L/home/haris/Diplwmtikh/affix-2.0.2/sdp/.libs
-L/home/haris/Diplwmtikh/affix-2.0.2/lib/.libs -L/usr/local/lib -
laffix_sdp -laffix_sdpsvc -laffix -L/usr/lib -lopenobex -Wl,-
soname -Wl,libaffix_obex.so.2 -o .libs/libaffix_obex.so.2.0.2
/usr/bin/install -c .libs/libaffix_obex.so.2.0.2T
/usr/local/lib/libaffix_obex.so.2.0.2
(cd /usr/local/lib && rm -f libaffix_obex.so.2 && ln -s
libaffix_obex.so.2.0.2 libaffix_obex.so.2)
(cd /usr/local/lib && rm -f libaffix_obex.so && ln -s
libaffix_obex.so.2.0.2 libaffix_obex.so)
/usr/bin/install -c .libs/libaffix_obex.lai
/usr/local/lib/libaffix_obex.la
/usr/bin/install -c .libs/libaffix_obex.a
/usr/local/lib/libaffix_obex.a
ranlib /usr/local/lib/libaffix_obex.a
chmod 644 /usr/local/lib/libaffix_obex.a
PATH="$PATH:/sbin" ldconfig -n /usr/local/lib
-----
-----

```

```

Libraries have been installed in:
    /usr/local/lib

```

If you ever happen to want to link against installed libraries in a given directory, LIBDIR, you must either use libtool, and specify the full pathname of the library, or use the `-LLIBDIR'

```

flag during linking and do at least one of the following:
- add LIBDIR to the `LD_LIBRARY_PATH' environment variable
  during execution
- add LIBDIR to the `LD_RUN_PATH' environment variable
  during linking
- use the `-Wl,--rpath -Wl,LIBDIR' linker flag
- have your system administrator add LIBDIR to
  `/etc/ld.so.conf'

```

See any operating system documentation about shared libraries for more information, such as the ld(1) and ld.so(8) manual pages.

```

-----
-----

```

```

/bin/sh ../mkinstalldirs /usr/local/bin
/bin/sh ../libtool --mode=install /usr/bin/install -c btobex
/usr/local/bin/btobex
/usr/bin/install -c .libs/btobex /usr/local/bin/btobex
make[2]: Nothing to be done for `install-data-am'.
make[2]: Leaving directory `/home/haris/Diplwmtikh/affix-
2.0.2/obex'
make[1]: Leaving directory `/home/haris/Diplwmtikh/affix-
2.0.2/obex'
Making install in tools
make[1]: Entering directory `/home/haris/Diplwmtikh/affix-
2.0.2/tools'
make[2]: Entering directory `/home/haris/Diplwmtikh/affix-
2.0.2/tools'
/bin/sh ../mkinstalldirs /usr/local/bin
/bin/sh ../libtool --mode=install /usr/bin/install -c btmodem
/usr/local/bin/btmodem
/usr/bin/install -c .libs/btmodem /usr/local/bin/btmodem
make[2]: Nothing to be done for `install-data-am'.
make[2]: Leaving directory `/home/haris/Diplwmtikh/affix-
2.0.2/tools'
make[1]: Leaving directory `/home/haris/Diplwmtikh/affix-
2.0.2/tools'
Making install in daemon
make[1]: Entering directory `/home/haris/Diplwmtikh/affix-
2.0.2/daemon'
make[2]: Entering directory `/home/haris/Diplwmtikh/affix-
2.0.2/daemon'
/bin/sh ../mkinstalldirs /usr/local/bin
/bin/sh ../libtool --mode=install /usr/bin/install -c btsrv
/usr/local/bin/btsrv
/usr/bin/install -c .libs/btsrv /usr/local/bin/btsrv
make[2]: Nothing to be done for `install-data-am'.
make[2]: Leaving directory `/home/haris/Diplwmtikh/affix-
2.0.2/daemon'
make[1]: Leaving directory `/home/haris/Diplwmtikh/affix-
2.0.2/daemon'
Making install in control
make[1]: Entering directory `/home/haris/Diplwmtikh/affix-
2.0.2/control'
make[2]: Entering directory `/home/haris/Diplwmtikh/affix-
2.0.2/control'
/bin/sh ../mkinstalldirs /usr/local/bin
/bin/sh ../libtool --mode=install /usr/bin/install -c btctl
/usr/local/bin/btctl
/usr/bin/install -c .libs/btctl /usr/local/bin/btctl
make install-exec-hook

```

```

make[3]: Entering directory `/home/haris/Diplwmtikh/affix-
2.0.2/control'
ln -sf /usr/local/bin/btctl /usr/local/bin/btftp
make[3]: Leaving directory `/home/haris/Diplwmtikh/affix-
2.0.2/control'
make[2]: Nothing to be done for `install-data-am'.
make[2]: Leaving directory `/home/haris/Diplwmtikh/affix-
2.0.2/control'
make[1]: Leaving directory `/home/haris/Diplwmtikh/affix-
2.0.2/control'
Making install in etc
make[1]: Entering directory `/home/haris/Diplwmtikh/affix-
2.0.2/etc'
make[2]: Entering directory `/home/haris/Diplwmtikh/affix-
2.0.2/etc'
if [ -f /etc/debian_version ] ; then \
    cp affix.rc.deb /etc/init.d/affix || true; \
    if [ "" = "" ] ; then \
        update-rc.d affix defaults; \
    fi \
fi
if [ -f /etc/redhat-release ] ; then \
    cp affix.rc.rh /etc/rc.d/init.d/affix || true; \
    if [ "" = "" ] ; then \
        chkconfig --add affix; \
    fi \
fi
/bin/sh ../mkinstalldirs /etc/affix
/usr/bin/install -c ppp_server /etc/affix/ppp_server
/usr/bin/install -c serial /etc/affix/serial
/usr/bin/install -c btsrv-gui /etc/affix/btsrv-gui
/usr/bin/install -c affix /etc/affix/affix
/bin/sh ../mkinstalldirs /etc/ppp/ip-down.d
mkdir /etc/ppp/ip-down.d
/usr/bin/install -c masq-down /etc/ppp/ip-down.d/masq-down
/bin/sh ../mkinstalldirs /etc/ppp/ip-up.d
mkdir /etc/ppp/ip-up.d
/usr/bin/install -c masq-up /etc/ppp/ip-up.d/masq-up
/bin/sh ../mkinstalldirs /etc/affix
/usr/bin/install -c -m 644 ./btsrv.conf /etc/affix/btsrv.conf
/usr/bin/install -c -m 644 ./affix.conf /etc/affix/affix.conf
/usr/bin/install -c -m 644 ./device.map /etc/affix/device.map
make[2]: Leaving directory `/home/haris/Diplwmtikh/affix-
2.0.2/etc'
make[1]: Leaving directory `/home/haris/Diplwmtikh/affix-
2.0.2/etc'
Making install in include
make[1]: Entering directory `/home/haris/Diplwmtikh/affix-
2.0.2/include'
make[2]: Entering directory `/home/haris/Diplwmtikh/affix-
2.0.2/include'
make[2]: Nothing to be done for `install-exec-am'.
/bin/sh ../mkinstalldirs /usr/include/affix
/usr/bin/install -c -m 644 btcore.h /usr/include/affix/btcore.h
/usr/bin/install -c -m 644 utils.h /usr/include/affix/utils.h
/usr/bin/install -c -m 644 sdp.h /usr/include/affix/sdp.h
/usr/bin/install -c -m 644 sdpclt.h /usr/include/affix/sdpclt.h
/usr/bin/install -c -m 644 sdpsrv.h /usr/include/affix/sdpsrv.h
/usr/bin/install -c -m 644 obex.h /usr/include/affix/obex.h
make[2]: Leaving directory `/home/haris/Diplwmtikh/affix-

```



```
2.0.2/include'
make[1]: Leaving directory `/home/haris/Diplwmtikh/affix-
2.0.2/include'
make[1]: Entering directory `/home/haris/Diplwmtikh/affix-2.0.2'
make[2]: Entering directory `/home/haris/Diplwmtikh/affix-2.0.2'
# PROGS and LIBSS
make[2]: Nothing to be done for `install-data-am'.
make[2]: Leaving directory `/home/haris/Diplwmtikh/affix-2.0.2'
make[1]: Leaving directory `/home/haris/Diplwmtikh/affix-2.0.2'
[root@Haris affix-2.0.2]#
```

8 Σύνδεση USB Bluetooth συσκευής

Όπως έχει αναφερθεί, για συσκευές USB Bluetooth συνιστάται η εγκατάσταση του hotplug πακέτου. Το hotplug σου δίνει τη δυνατότητα, με το που συνδέσεις το usb Bluetooth στον υπολογιστή, να το χρησιμοποιήσεις αμέσως. Το απαραίτητο module btusb.o θα φορτωθεί αυτόματα αν χρειαστεί. Αν για τον οποιοδήποτε λόγο αυτό δεν συμβεί, θα χρειαστεί να το κάνετε αυτό χειροκίνητα.

8.1 Εντολές

8.1.1 Αναζήτηση γειτονικών συσκευών Bluetooth

Υπάρχουν δύο βασικές εντολές που βοηθάνε στον εντοπισμό.

Η btctl discovery και η btctl inquiry. Και οι δύο εντοπίζουν γειτονικές συσκευές Bluetooth μόνο που η discovery πέρα από τις διευθύνσεις και το είδος των συσκευών που βρέθηκαν μας δίνει και τα ονόματα που αυτές έχουν κάτι που δεν το κάνει η inquiry, αλλά η discovery είναι σαφώς πιο αργή.

● btctl discovery [length]

Η οποία ψάχνει για length χρόνο να βρει γειτονικές συσκευές Bluetooth. Αν δεν δοθεί το όρισμα length τότε σαν χρόνος αναζήτησης ορίζονται τα 8sec. Το ίδιο και για την btctl inquiry [length].

Τυπική έξοδος μιας btctl discovery εντολής δίνεται παρακάτω.

```
haris@alexis:~$ btctl discovery 5
Searching ...
Searching done. Resolving names ...
done.
+1: Address: 00:02:ee:46:22:39, Class: 0x502204, Name: "Nokia 3650"
Phone (Cellular) [Object Transfer,Telephony]
+2: Address: 00:e0:03:74:4e:3d, Class: 0x56010C, Name: "Windows 2k"
```

Όπως βλέπουμε παραπάνω, κάθε εντοπισμένη συσκευή, έχει στα αριστερά της έναν αριθμό. Ο αριθμός αυτός, μπορεί να χρησιμοποιηθεί στη συνέχεια στη θέση της διεύθυνσης.

8.1.2 Εντοπισμός διαθέσιμων υπηρεσιών σε μια εντοπισμένη Bluetooth συσκευή

Κάθε bluetooth συσκευή έχει έναν αριθμό από διαθέσιμες υπηρεσίες που μπορεί να χρησιμοποιήσει μια άλλη συσκευή. Για να βρεθούν οι διαθέσιμες υπηρεσίες της απομακρυσμένης συσκευής χρειάζεται να υπάρχει υποστήριξη από το SDP πρωτόκολλο (Service Discovery Protocol). Το πρωτόκολλο αυτό, το παρέχει το affix από μόνο του.

Για να δούμε τις διαθέσιμες υπηρεσίες μιας απομακρυσμένης συσκευής, χρησιμοποιούμε τις εντολές: *btctl browse <address>* και η *btctl search <address>*

Όπου <address> η διεύθυνση ή ο αριθμός της συσκευής. Πρέπει να πούμε πως μερικές συσκευές δεν υποστηρίζουν την εντολή browse.

Τυπική έξοδος μιας btctl browse εντολής δίνεται παρακάτω.

```
haris@alexis:~$ btctl browse 1
Connecting to host 00:02:ee:46:22:39 ...
=====
Service Name: Fax
-----
II
Chapter 5. Usage and Features
SvcRecHdl: 0x10000
Service Class ID List:
"Fax" (0x1111)
```

```

"Generic Telephony" (0x1204)
Protocol Descriptor List:
"L2CAP" (0x0100)
"RFCOMM" (0x0003)
Port/Channel: 1
Profile Descriptor List:
"Fax" (0x1111)
Version: 0x0100
Browse Group List:
"PublicBrowseGroup" (0x1002)
=====
Service Name: Dial-up Networking
-----
... skipped
=====
Service Name: Bluetooth Serial Port
-----
... skipped
=====
Service Name: OBEX Object Push
-----
... skipped
=====
Service Name: OBEX File Transfer
-----
... skipped

```

8.1.3 Σύνδεση με μια Bluetooth συσκευή

Για λόγους ασφαλείας, δηλαδή για να μην μπορεί ο καθένας να συνδέεται σε μια συσκευή χωρίς την άδεια του κατόχου, δίνουμε ένα password στη συσκευή μας ώστε με αυτό τον τρόπο, να μπορούν να συνδεόνται μόνο αυτοί που επιθυμούμαι. Όσοι δεν ξέρουν το password, δεν θα έχουν και τη δυνατότητα για μια σύνδεση.

```
haris@alexis:~$ btctl addpin default "1234"
```

Με αυτό τον τρόπο, θέτουμε by default κωδικό, που ισχύει για όλες τις συνδέσεις. Αν αντί για default βάζαμε το νούμερο ή τη διεύθυνση μιας συσκευής, τότε αυτό το pin θα ίσχυε μόνο για εκείνη τη συσκευή.

8.1.4 Θέτοντας μία PPP Connection

Σύνδεση μέσω μια DUN (Dial up Networking) σύνδεσης.

```
haris@alexis:~$ btctl connect 1 DUN
Connecting to host 00:02:ee:46:22:39 ...
Service found on channel 1
Connecting to channel 1 ...
Connected. Bound to line 0 [/dev/bty0].
```

Τώρα η σύνδεση έχει επιτευχθεί και το τηλέφωνο, δουλεύει σαν modem. Το μόνο που πρέπει να ρυθμίσεις είναι να αλλάξεις το όνομα της συσκευής στο `ppp configuration` αρχείο από `/dev/ttySx` σε `/dev/btyx`.

8.1.5 Χρησιμοποιώντας το OBEX για μεταφορά αρχείων από και προς το κινητό

Το OBEX πρωτόκολλο (Object Exchange protocol) καλύτερα μπορεί να περιγραφεί σαν δυαδικό HTTP. Το OBEX μπορεί να χρησιμοποιηθεί για την ανταλλαγή οποιουδήποτε αρχείου όπως φωτογραφίες, αρχεία δεδομένων και άλλα.

Το OBEX υπάρχει σε συσκευές όπως είναι τα PDA's όπως είναι το Palm Pilot και στα κινητά τηλέφωνα της NOKIA όπως είναι το 7650, το 3650 και άλλα. Για να στείλουμε ένα αρχείο από το κινητό, χρειάζεται να έχουμε τρέξει την εντολή `btsrv`. Δηλαδή να σηκώσουμε το server (Bluetooth server). Όλα τα αρχεία που στέλνουμε από το κινητό αποθηκεύονται σε ένα προκαθορισμένο φάκελο με το όνομα `/Inbox`.

Παρακάτω δείχνουμε με πιο τρόπο στέλνουμε μια εικόνα σε ένα κινητό.

```
haris@alexis:~$ btctl push 1 haris.jpg
Transfer complete.
28998 bytes sent in 9.38 secs (3091.47 B/s)
```

9 Ρυθμίσεις κινητού(Τελικό Στάδιο)

Αφού έχουμε ολοκληρώσει επιτυχώς τα παραπάνω στάδια, προχωράμε στο τελικό στάδιο που αφορά τις ρυθμίσεις που πρέπει να γίνουν στο κινητό και τις τελικές ρυθμίσεις στον υπολογιστή ώστε να φτάσουμε στο επιθυμητό αποτέλεσμα.

Πρώτα από όλα, πρέπει να εγκαταστήσουμε μια εφαρμογή στο κινητό που ονομάζεται GNUBOX. Η επόμενη παράγραφος, δίνει κάποιες βασικές πληροφορίες για το GNUBOX, τι είναι, πως δουλεύει, πως θα το εγκαταστήσουμε στο κινητό και ποιες ρυθμίσεις πρέπει να κάνουμε.

9.1 GNUBOX

Το Gnuibox είναι ένα απλό πρόγραμμα που αναπτύχθηκε για την σειρά 60 έκδοσης της Nokia (3650, 7650,...) το οποίο επιτρέπει στο κινητό να συνδεθεί με το διαδίκτυο μέσα από ένα σημείο πρόσβασης των υπερύθρων ή του Bluetooth. Με άλλα λόγια σου επιτρέπει να έχεις IP συνδεσιμότητα χωρίς να χρησιμοποιείς την πρωταρχική στοίβα τηλεφώνου(CSD or GPRS). Οι δυνατότητες αυτού του προγράμματος είναι απεριόριστες αφού προσφέρει στο χρήστη μια περιήγηση στο διαδίκτυο χωρίς να επιβαρύνεται οικονομικά, και με ταχύτητες μεγαλύτερες από αυτές που προσφέρει μια απλή σύνδεση μέσω GPRS ή άλλες μορφές πρόσβασης στο internet μέσω του κινητού τηλεφώνου του.

Το Gnuibox μπορείτε να το βρείτε στην παρακάτω διεύθυνση :

<http://www.symbianos.org/cgi-bin/viewcvs.cgi/gnuibox/>

Η εγκατάσταση του είναι αρκετά εύκολη αφού απλά εγκαθιστάς το αρχείο gnuibox.sis* στο κινητό και ακολουθείς την εξής διαδικασία:

Menu → Εργαλεία → Ρυθμίσεις → Σύνδεση → Σημεία πρόσβασης → Επιλογές → Νέο σημείο πρόσβασης → Χρήση προεπιλ.ρυθμ.

Όνομα σύνδεσης

Τυχαίο

Φορέας δεδομένων

Δεδ. GSM

Αριθμός σύνδεσης

Κενό

Όνομα χρήστη

Κενό

Προτροπή κωδ.πρόσβασης

Όχι

Κωδικός πρόσβασης

τυχαίος

Έλεγχος στοιχείων

ασφαλής

Διεύθυνση πύλης IP

0.0.0.0

Τύπος κλήσεων δεδομένων

Αναλογική

Μέγιστη ταχύτητα δεδομένων

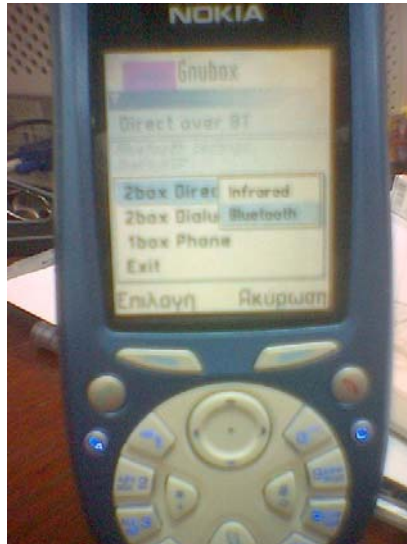
Αυτόματη

Επίσης πρέπει στο πεδίο Ρυθμίσεις σύνδεσης → GPRS να οριστεί

Σύνδεση GPR

Όταν απαιτείται και δεν είναι ήδη προεπιλεγμένο.

* Μπορείς να στείλεις το gnuBox.sis μέσω Bluetooth στο κινητό με χρήση της εντολής btctl που αναφέραμε πιο πάνω.



Στη συνέχεια ενεργοποιούμε έναν πρόγραμμα περιήγησης στο διαδίκτυο π.χ Opera ή έναν Videoplayer. Στην εκπόνηση της παρούσας διπλωματικής εργασίας χρησιμοποιήσαμε έναν player και συγκεκριμένα τον RealOne Player 3.1.234. Ο τρόπος χρήσης του Player είναι αρκετά εύκολος παρόλα αυτά, εμείς θα αναφέρουμε το βασικό τρόπο χρήσης.

RealOne Player → Άνοιγμα → Διεύθυνση URL(**Πληκτρολογείται η διεύθυνση του ζητούμενου videoclip**)→ **Επιλέγεται το όνομα της σύνδεσης που δημιουργήθηκε παραπάνω**

9.2 Ρυθμίσεις από την πλευρά του υπολογιστή

Για να θέσουμε τη σύνδεση από τη πλευρά του LINUX πρέπει να ετοιμάσουμε τον PPP daemon για εισερχόμενες συνδέσεις. Και για αυτή τη διαδικασία που θα περιγράψουμε παρακάτω, πρέπει να έχουμε κάνει login σαν χρήστης root ώστε να έχουμε πλήρης πρόσβαση στον υπολογιστή. Υπάρχουν δύο τρόποι για να προχωρήσουμε. Ο πρώτος είναι να αλλάξουμε τις ρυθμίσεις στο αρχείο /etc/ppp/options, αλλά είναι καλύτερο να προχωρήσουμε με τον δεύτερο τρόπο να στήσουμε τις εισερχόμενες pppd συνδέσεις με ξεχωριστά pppd peer έγγραφο (scripts).

Πρώτα, πρέπει να δημιουργήσουμε ένα φάκελο στο μονοπάτι /etc/ppp/peers (π.χ /etc/ppp/peers/myBluetooth3650) το οποίο, θα περιέχει πληροφορίες για τη σειριακή θύρα σε εισερχόμενη υπηρεσία rpprd και επιπλέον πληροφορίες για την πιστοποίηση της σύνδεσης, DNS και IP ρυθμίσεις:

```
noauth
local
modem
proxyarp
/dev/bty0
ms-dns <mydnsserveripaddress>
192.168.0.5 (IP Υπολογιστή) :192.168.0.6 (IP κινητού)
```

Μετά πρέπει να καλέσουμε τον rpprd daemon με την εντολή:

```
➤ rpprd -d -detach call myBluetooth7650
```

Για να πετύχουμε τη σωστή λειτουργία του GNUBOX, χρειάζεται να τρέξουμε την εντολή:

```
➤ echo "1" > /proc/sys/net/ipv4/ip_forward
```



9.3 Συμπεράσματα

Η σύνδεση με το διαδίκτυο ήταν επιτυχής και με αποθηκευμένο videoclip καθώς και με streaming video. Καθόλη τη διάρκεια των πειραμάτων ο χρήστης δεν αντιμετώπισε προβλήματα με την λήψη όσο βρισκόταν εντός του πεδίου κάλυψης του Bluetooth, ενώ μπορούσε παράλληλα να εκτελέσει και άλλες υπηρεσίες όπως να διεκπεραιώσει τηλεφωνικές συνδιαλέξεις και ανταλλαγή μηνυμάτων SMS

Βιβλιογραφία

➡ Internet

- [1]<http://www.cs.utk.edu/~dasgupta/bluetooth/history.htm> Bluetooth Protocol and security architecture review. Project submitted for computer network security by Korak Dasgupta
- [2]<http://linux10.org/history/> San Jose, Berkeley
- [3]<http://workshop.molecularrevolution.org/resources/computing.php> by The Marine Biological Laboratory.
- [4]<http://www.tamos.net/ieee/history.html> by IEEE Computer Society.
- [5]<http://www.linux.org/info/> by Linux Online Inc.
- [6]<http://bthow.sourceforge.net/> by Andrei Emeltchenko developer in Nokia research Center at Mobile Network
- [7]<http://affix.sourceforge.net/> by Dmitry Kasatkin, Carlos Chinae developers in Nokia research Center at Mobile Network
- [8]<http://affix.sourceforge.net/affix-doc/index.html> by Dmitry Kasatkin developer in Nokia research Center at Mobile Network.
- [9] <http://affix.sourceforge.net/affix-newdoc/Affix-enduser/index.html> by Henri Ranky Developer in Nokia.
- [10]<http://affix.sourceforge.net/hardware.shtml> by Andrei Emeltchenko (idea came from Marcel Holtmann (marcel@holtmann.org))
- [11]<http://www.aboutdebian.com/> by Keith Parkansky.
- [12]<http://www.suso.org/linux/tutorials/ssh.phtml> SSH Tutorial for Linux by Suso Banderas.
- [13]<http://www.cs.uchicago.edu/docs/networking/ssh> by the University of Chicago Department of Computer Science.
- [14]<http://symbianos.org/cgi-bin/viewcvs.cgi/gnubox/> by Mal Minhas
- [15]<http://symbianos.org/> by Mal Minhas
- [16]<http://symbianos.org/~malm/SymbianLinuxHowTo.html> by Mal Minhas
- [17]<http://www.holtmann.org/linux/bluetooth/devices.html> by Marcel Holtmann, Holtmann Organization.
- [18]<http://openobex.sourceforge.net>
- [19]<http://www.forum.nokia.com> official website of Nokia Corporation
- [20]<http://www.bluetooth.com> official website of Bluetooth

➡ Mailing Lists

- [21]<mailto:affix-devel@lists.sourceforge.net>
- [22]<http://www.symbianos.org/>
- [23]<http://sourceforge.net/mailarchive/forum.php?forum=affix-devel>

➡ Pdf Αρχεία

- [24]howto.pdf Affix Bluetooth HOWTO by Andrei Emeltchenko at Nokia.com
- [25]Basic Linux Commands.pdf
- [26]BasicUnixCommands.pdf
- [27]INFO-SHEET.pdf Linux Information Sheet by Michael K. Johnson

- [28]intro-linux.pdf Introduction to Linux by Machtelt Garrels
- [29]linux.pdf Introduction to Linux by Mark Earnest Emerging Technologies Information Technology Services
- [30]linux-windows.pdf An examination of the purchase and total operational costs of running an enterprise on Linux/Open Source in comparison to Microsoft's Windows computer system platforms by Cybersource Pty. Ltd.
- [31]What is Linux.pdf What is Linux?By Gavin Henry Suretec Systems Ltd.
- [32]Bluetooth.pdf The Bluetooth Technology by Pankaj Bhatt
- [33]BluetoothΕφαρμογές.pdf
- [34]Introducing Bluetooth.pdf
- [35]Ανάπτυξη εφαρμογής εξυπηρετητή (σε Linux) – πελάτη (σε Symbian) για την μεταφορά και αναπαραγωγή αρχείων video σε έξυπνες συσκευές κινητής τηλεφωνίας πάνω από Bluetooth.

