Bluetooth

Vlad Manea :: *vlad.manea@info.uaic.ro*

Bluetooth este un protocol pentru schimbul de date fără fir pe distanţe scurte ce foloseşte unde scurte radio şi este utilizat de dispozitive fixe şi mobile. Prin intermediul Bluetooth, se creează reţele de acoperire mică de tip PAN. Acest protocol a fost iniţial conceput ca o alternativă fără fir la conexiunile prin cablu. Prin bluetooth se pot conecta mai multe dispozitive, depăşindu-se problemele de sincronizare.

Logo-ul Bluetooth
reuneşte caracterele

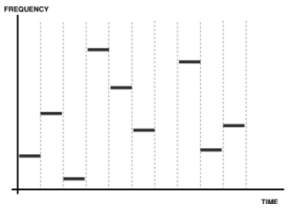
şi .

Denumire

Bluetooth este versiunea engleză a cuvântului Blatand, epitet asociat regelui Herald I al Danemarcei şi Norvegiei care a unit triburile nordice într-un regat. Bluetooth face acelaşi lucru cu protocoalele de comunicaţii, şi anume le uneşte într-un standard universal.

Implementare

Bluetooth foloseste o tehnologie radio numită spectru de răspândire cu salt al frecvenţei – Frequency-Hopping Spread Spectrum, FHSS – care împarte datele trimise şi transmite părţi din acestea pe un număr de până la 79 frecvenţe. Rata de transmisie poate ajunge până la 1 Mb/s. Bluetooth oferă un mod de a conecta şi schimba informaţii între dispozitive, cum ar fi: telefoanele mobile, calculatoarele portabile, calculatoarele personale, imprimantele, sisteme de poziţionare globală prin satelit, GPS, receptoarele, camerele digitale, precum şi consolele de jocuri video printr-o bandă sigură, de rază scurtă de acţiune, de tipul ISM – Industrial, Scientific and Medical – la frecvenţa de 2.4 GHz. Specificaţiile Bluetooth sunt dezvoltate şi licenţiate de Bluetooth Special Interest Group, SIG, format din societăţi ce activează în domeniile: telecomunicaţii, informatică, reţele, precum şi electronice de consum.



Saltul frecvenţei
în timp, frecvenţa
ia valori diferite

Utilizări

Bluetooth este un standard şi un protocol de comunicaţii în primul rând proiectat pentru consum redus de energie, cu o rază scurtă de acţiune, de puteri clasificate după distanţa maximă: 100m, 10m şi 1m, pe baza unor cipuri de costuri reduse în fiecare dispozitiv. Bluetooth face posibil ca aceste dispozitive să comunice între ele atunci când sunt în aria de acoperire. Deoarece dispozitivele folosesc un sistem de comunicare prin unde radio – broadcast, ele nu trebuie să fie în aceeaşi cameră.

Clasă	Putere (mW)	Rază (m)
1	100	100
2	2.5	22
3	1	6

Clasele putere-rază
permit costuri reduse

Gama de utilizări ale Bluetooth este largă:

- comunicarea între un telefon mobil şi o cască hands-free
- reţea fără fir între calculatoare într-un spaţiu limitat şi lăţime de bandă mică
- comunicarea dintre calculatoare şi periferice: mouse, tastatură, imprimantă
- comunicarea dispozitivelor medicale
- comunicarea scannerelor de coduri de bare
- comunicarea dispozitivelor de control al traficului
- comunicarea când lărgimea de bandă USB nu este necesară şi nu se vrea fir
- comunicarea cu joystick-urile pentru console: Nintendo Wii, PlayStation 3
- comunicarea dintre un telefon capabil de conexiune la internet şi un calculator sau PDA

Cască Bluetooth
Nokia BH-209

Bluetooth versus Wi-Fi

Dispozitiv Wi-Fi
Linksys WRT600N

Bluetooth şi Wi-Fi Ethernet IEEE 802.11 au multe aplicaţii în birourile de astăzi şi acasă: crearea de reţele, imprimarea sau transferul de fişiere de la PDA-uri la calculatoare.

Wi-Fi este destinat pentru echipamente de rezidenţi şi aplicaţiile lor. Această categorie este încadrată ca WLAN (reţea locală fără fir). Wi-Fi este conceput ca un înlocuitor pentru cablarea din zona de reţele locale la locul de muncă.

Bluetooth este destinat pentru echipamente de nerezidenţi şi aplicaţiile lor. Categoria este încadrată la WPAN. Bluetooth înlocuieşte cablul în apropierea unei persoane.

Cerinţe

Adaptor Bluetooth
Star Max BT15

Calculatorul personal trebuie să aibă un adaptor pentru a comunica prin Bluetooth cu alte dispozitive. Cele mai recente calculatoare sunt dotate cu un astfel de adaptor. Spre deosebire de predecesorul său, IrDA, care necesită câte un adaptor pentru fiecare dispozitiv, Bluetooth permite comunicarea unor dispozitive multiple cu un calculator printr-un singur adaptor.

Apple a permis Bluetooth de la sistemul de operare Mac OS X 10.2, lansat în 2002. Pentru platformele Microsoft, Windows XP Service Pack 2 şi sistemele de operare ulterioare suportă Bluetooth. Linux are implementările proprii pentru Bluetooth, sub numele BlueZ şi Affix.

Un telefon mobil cu opţiunea Bluetooth trebuie să respecte recomandările documentului Bluetooth Local Connectivity, articol redactat de Open Mobile Terminal Platform (OMTP).

Versiuni

Specificaţiile Bluetooth au fost realizate în 1994 de suedezii Japp Haartsen şi Sven Mattisson şi formalizate de Bluetooth Special Interest Group (SIG), înfiinţat în 1998. În momentul de faţă, numărul de membri depăşeşte 11.000 companii, printre care se află: IBM, Intel, Toshiba, Nokia, Sony.

Versiuni Bluetooth:

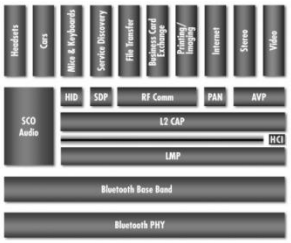
- **1.0** şi **1.0B** au avut probleme, iar producătorii au avut dificultăţi de a face dispozitivele interoperabile. Versiunile cereau transmiterea adreselor hardware ale dispozitivelor, făcând anonimitatea imposibilă.
- **1.1** a fost ratificată ca IEEE 802.15 în 2002, a fost adăugat un indicator pentru puterea semnalului primit şi s-a permis comunicarea prin canale necriptate.
- **1.2** a fost ratificată ca IEEE 802.15 în 2005 şi este compatibilă cu 1.1 şi permite o conexiune mai rapidă, rezistenţă la interferenţa radio, viteze mai mari de transmisie, de până la 721 Kb/s, calitate îmbunătăţită a sunetului prin retransmiterea pachetelor stricate.
- **2.0 + EDR** a fost lansată în 2004, este compatibilă cu 1.2, iar diferenţa semnificativă constă în viteza de transfer a datelor până la 2.1 Mb/s în practică, la un consum redus prin reducerea ciclului de funcţionare. Unele produse permit Bluetooth 2.0, ceea ce nu include rata de transfer a datelor îmbunătăţită
- **2.1 + EDR** este compatibilă cu 1.2 şi a fost adoptată de SIG în 2007 şi suportă viteze teoretice de până la 3 Mb/s.
- **3.0 + HS** a fost adoptată de SIG în 2009 şi suportă, teoretic, 24 Mb/s.

Japp Haartsen
inventator Bluetooth

Sven Mattisson
inventator Bluetooth

Protocoale

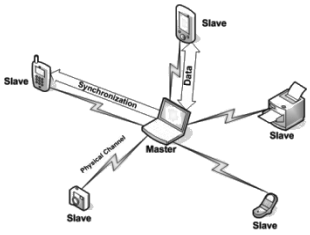
Bluetooth este definit ca o arhitectură de stivă de protocoale ce conţine protocoale de bază, protocoale de înlocuire a cablului, protocoale de control al telefoniei şi protocoale adiptate. Protocoalele obligatorii pentru toate stivele Bluetooth sunt: LMP, L2CAP şi SDP. Adiţional, aceste protocoale sunt suportate: HCI şi RFCOMM.



Stiva de protocoale
partea superioară:
- căşti
- automobile
- mouse şi tastatură
- descoperirea serviciilor
- transfer de fişier
- schimburi card
- imprimare
- fotografiere
- internet
- stereo
- video

- **Protocolul de dirijare a legăturilor LMP** este folosit pentru controlul legăturii radio dintre două dispozitive.
- **Protocolul de control logic al legăturilor şi adaptare L2CAP** este folosit pentru multiplexarea conexiunilor logice multiple între două dispozitive folosind protocoale de nivel mai înalt. Implementează segmentarea şi reasamblarea pachetelor în timp real. În modul de bază, permite pachete de dimensiune până la 64 KB.
- **Protocolul de descoperire a serviciilor SDP** permite dispozitivelor să afle ce servicii suportă celelalte dispozitive şi ce parametri sunt necesari pentru conectarea la acestea. De exemplu, când se conectează un mobil la un set de căşti, SDP va fi utilizat pentru a determina ce profiluri sunt suportate de set şi setările necesare pentru conectarea pentru fiecare profil. Fiecare serviciu este identificat printr-un identificator unic universal, UUID.
- **Interfaţa dintre gazdă şi controlor HCI** este un protocol de comunicare standard între stiva gazdă, de exemplu, sistemul de operare al unui calculator şi controlorul Bluetooth. Acest standard permite stivei gazdă să fie interschimbată cu un minim de adaptare.
- **Protocolul de înlocuire a cablului RFCOMM** – Radio Frequency COMMunications – creează un flux serial virtual de date. Oferă un flux de date fiabil pentru utilizator, cum e TCP. Este larg utilizat datorită API-urilor disponibile pe majoritatea sistemelor de operare.

Canal fizic

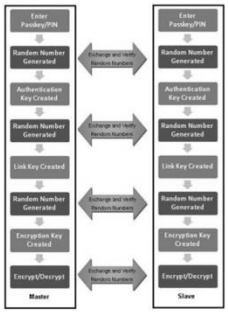


Modelul Master-Slave
între master şi fiecare slave:
- transmisie de date
- sincronizare

Nivelul fizic Bluetooth (RF) operează în banda nelicenţiată ISM la 2.4 GHz. În timpul operaţiei tipice, un canal radio fizic este împărţit de un grup de până la 8 dispozitive sincronizate la un ceas şi pe un pattern de frevenţă comune. Un dispozitiv ia rolul de referinţă şi este cunoscut ca master. Celelalte dispozitive sunt numite slave. Un astfel de grup formează un piconet. Aceasta este forma de bază de comunicare pentru tehnologia fără fir Bluetooth.

Aceste dispozitive folosesc un pattern pentru frecvenţă determinat algoritmic conform ceasului. Pattern-ul de bază este o ordonare pseudo-aleatoare a celor 79 de frecvenţe de pe banda ISM. Acest pattern poate fi adaptat să excludă o porţiune a frecvenţelor utilizate de dispozitivele ce interferă.

Canalul fizic este divizta în unităţi de timp numite şi sloturi. Datele sunt transmise între dispozitive Bluetooth în pachete poziţionate în aceste sloturi. Când circumstanţele permit, un număr de sloturi consecutive pot fi alocate unui singur pachet. Tehnologia Bluetooth permite, astfel, transmisia full-duplex prin această schemă.



Împerecherea Master-Slave
generare, verificare de nr. random
primele 3 niveluri: împerechere
ultimul nivel: sesiune

Împerechere

Multe servicii Bluetooth pot expune date private sau pot permite partea conectată să controleze dispozitivul Bluetooth. Din motive de securitate, e necesar controlul căror dispozitive le este permis să se conecteze la un dispozitiv Bluetooth. În acelaşi timp, este util ca dispozitivele să stabilească în mod automat o conexiune când sunt în aria de acoperire.

Pentru a rezolva acest conflict, două dispozitive trebuie să fie împerecheate pentru a comunica unul cu celălalt. Procesul de împerechere este, de obcei, declanşat automat când primeşte o primă cerere de conexiune de la un dispozitiv cu care nu este împerecheat. După ce împerecherea a fost stabilită, ea este reţinută de dispozitive, care se pot conecta între ele fără intervenţia utilizatorului. Când se vrea, relaţia de împerechere poate fi eliminată de utilizator.

Securitate

Bluetooth implementează confidenţialitatea, autentificarea şi derivarea cheilor cu algoritmi ce se bazează pe codul bloc SAFER+. În Bluetooth, generarea cheilor se bazează în general pe un PIN, care trebuie introdus în ambele dispozitive. Această procedură poate, bineînţeles, fi modificată dacă un dispozitiv are o interfaţă limitată sau un PIN fix: de exemplu, căştile.

you have just been bluejacked ☺

Posibil mesaj bluejack inofensiv

În timpul împerecherii, o cheie de iniţializare sau cheie master este generată. Codul E0 este utilizat pentru criptarea pachetelor, garantarea confidenţialităţii şi se bazează pe un secret criptografic împărţit, în speţă o cheie master generată anterior. Aceste chei, utilizate pentru repetate criptări ale datelor trimise se bazează pe codurile PIN introduse în unul sau ambele dispozitive conectate.

Atacul Bluejacking este trimiterea unei fotografii sau a unui mesaj de către un utilizator unui altul prin tehnologia fără fir Bluetooth. Aplicaţiile includ mesaje scurte care notifică utilizatorul cu privire la acest atac. Bluejacking nu are efecte asupra datelor reţinute în dispozitiv.

Mostră de cod

Se gestionează conexiunea în stiva de protocoale Bluetooth pentru Linux cu numele BlueZ. Codul prezentat este incomplet. Pentru codul complet, consultați bibliografia.

```
/*
 *
 *  BlueZ - Bluetooth protocol stack for Linux
 *
 *  Copyright (C) 2004-2009  Marcel Holtmann <marcel@holtmann.org>
 *
 *
 *  This program is free software; you can redistribute it and/or modify
 *  it under the terms of the GNU General Public License as published by
 *  the Free Software Foundation; either version 2 of the License, or
 *  (at your option) any later version.
 *
 *  This program is distributed in the hope that it will be useful,
 *  but WITHOUT ANY WARRANTY; without even the implied warranty of
 *  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 *  GNU General Public License for more details.
 *
 *  You should have received a copy of the GNU General Public License
 *  along with this program; if not, write to the Free Software
 *  Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA  02110-1301  USA
 *
 */

#ifdef HAVE_CONFIG_H
#include <config.h>
#endif

#include <stdio.h>
#include <errno.h>
#include <unistd.h>
#include <netinet/in.h>

#include <bluetooth/bluetooth.h>
#include <bluetooth/hci.h>
#include <bluetooth/bnep.h>
#include <bluetooth/sdp.h>

#include <glib.h>
#include <gdbus.h>

#include "logging.h"
#include "glib-helper.h"
#include "btio.h"
#include "dbus-common.h"
#include "adapter.h"
#include "device.h"

#include "error.h"
#include "common.h"
#include "connection.h"

#define NETWORK_PEER_INTERFACE "org.bluez.Network"

typedef enum {
        CONNECTED,
        CONNECTING,
        DISCONNECTED
} conn_state;

struct network_peer {
        bdaddr_t        src;
        bdaddr_t        dst;
        char            *path;          /* calea D-Bus */
        struct btd_device *device;
        GSList          *connections; };

struct network_conn {
        DBusMessage     *msg;
        char            dev[16];        /* Numele interfetei */
        uint16_t        id;             /* Rol: Identificator Clasa Serviciu */
        conn_state      state;
        GIOChannel      *io;
        guint           watch;          /* Deconectare */
        guint           dc_id;
        struct network_peer *peer; };
```

```
struct __service_16 {
	uint16_t dst;
	uint16_t src;
} __attribute__ ((packed));

static DBusConnection *connection = NULL;
static const char *prefix = NULL;
static GSList *peers = NULL;

/* Gaseste un pointer de tip struct network peer cu adresa data ca parametru in lista parametru */
static struct network_peer *find_peer(GSList *list, const char *path) {
	GSList *l;

	/* Itereaza printre peers */
	for (l = list; l; l = l->next) {
		struct network_peer *peer = l->data;
		/* L-am gasit! */
		if (!strcmp(peer->path, path))
			return peer; }
	/* Nu a gasit niciun peer */
	return NULL; }

/* Gaseste un pointer de tip network connection cu adresa data ca parametru in lista parametru */
static struct network_conn *find_connection(GSList *list, uint16_t id) {
	GSList *l;
	/* Itereaza printre conexiuni */
	for (l = list; l; l = l->next) {
		struct network_conn *nc = l->data;
		/* Am gasit o conexiune! */
		if (nc->id == id)
			return nc; }
	/* Nu a gasit nicio conexiune */
	return NULL; }

/* Functie de creare a erorii "nu este suportat" */
static inline DBusMessage *not_supported(DBusMessage *msg) {
	return g_dbus_create_error(msg, ERROR_INTERFACE ".Failed", "Not supported"); }

/* Functie de creare a erorii "deja conectat" */
static inline DBusMessage *already_connected(DBusMessage *msg) {
	return g_dbus_create_error(msg, ERROR_INTERFACE ".Failed", "Device already connected"); }

/* Functie de creare a erorii "neconectat" */
static inline DBusMessage *not_connected(DBusMessage *msg) {
	return g_dbus_create_error(msg, ERROR_INTERFACE ".Failed", "Device not connected"); }

/* Functie de creare a erorii "nu ai drepturi suficiente" */
static inline DBusMessage *not_permited(DBusMessage *msg) {
	return g_dbus_create_error(msg, ERROR_INTERFACE ".Failed", "Operation not permited"); }

/* Functie de creare a erorii "nu este suportat" */
static inline DBusMessage *connection_attempt_failed(DBusMessage *msg, const char *err) {
	return g_dbus_create_error(msg,ERROR_INTERFACE ".ConnectionAttemptFailed",
				err ? err : "Connection attempt failed"); }

/* Functie de revocare a conexiunii */
static void cancel_connection(struct network_conn *nc, const char *err_msg) {
	DBusMessage *reply;
	if (nc->watch) {
		g_dbus_remove_watch(connection, nc->watch);
		nc->watch = 0; }
	if (nc->msg && err_msg) {
		reply = connection_attempt_failed(nc->msg, err_msg);
		g_dbus_send_message(connection, reply); }
	/* Inchide canalul de comunicare, ii distruge referinte */
	g_io_channel_shutdown(nc->io, TRUE, NULL);
	g_io_channel_unref(nc->io);
	nc->io = NULL;

	/* Exista un automat finit, DISCONNECTED */
	nc->state = DISCONNECTED; }

/* Functie de distrugere a conexiunii */
static void connection_destroy(DBusConnection *conn, void *user_data) {
	struct network_conn *nc = user_data;
	if (nc->state == CONNECTED) {
		bnep_if_down(nc->dev);
		bnep_kill_connection(&nc->peer->dst); }
	else if (nc->io)
		cancel_connection(nc, NULL); }
```

```
/* Functie de conectare */
static int bnep_connect(struct network_conn *nc) {
        struct bnep_setup_conn_req *req;
        struct __service_16 *s;
        struct timeval timeo;
        unsigned char pkt[BNEP_MTU];
        int fd;

        /* Trimite cerere de conectare */
        req = (void *) pkt;
        req->type = BNEP_CONTROL;
        req->ctrl = BNEP_SETUP_CONN_REQ;
        req->uuid_size = 2;    /* UUID pe 16 biti */
        s = (void *) req->service;
        s->dst = htons(nc->id);
        s->src = htons(BNEP_SVC_PANU);

        memset(&timeo, 0, sizeof(timeo));
        timeo.tv_sec = 30;

        fd = g_io_channel_unix_get_fd(nc->io);
        setsockopt(fd, SOL_SOCKET, SO_RCVTIMEO, &timeo, sizeof(timeo));

        /* Pe aici am mai fost, stim ce se intampla ☺  */
        if (send(fd, pkt, sizeof(*req) + sizeof(*s), 0) < 0)
                return -errno;

        g_io_add_watch(nc->io, G_IO_IN | G_IO_ERR | G_IO_HUP | G_IO_NVAL,
                        (GIOFunc) bnep_setup_cb, nc);

        return 0; }

/* Functie de deconectare */
static DBusMessage *connection_disconnect(DBusConnection *conn, DBusMessage *msg, void *data) {
        struct network_peer *peer = data;
        GSList *l;

        /* Parcurg toate conexiunile */
        for (l = peer->connections; l; l = l->next) {
                struct network_conn *nc = l->data;
                if (nc->state == DISCONNECTED)
                        continue;
                /* Deconectez */
                return connection_cancel(conn, msg, nc); }

        /* Nu era nimeni conectat */
        return not_connected(msg); }
```

Bibliografie

- **Situl oficial al tehnologiei Bluetooth**
  http://www.bluetooth.com/Bluetooth/Technology/Works/
- **An Introduction to Spread Spectrum Techniques**, Carlo Kopp
  http://www.csse.monash.edu.au/~carlo/SYSTEMS/Spread-Spectrum-0597.htm
- **Wikipedia**
  http://en.wikipedia.org/wiki/Bluetooth
- **Situl oficial BlueZ**
  http://www.bluez.org