

O stație radio digitală ... “în buzunar”

Alexandru RADOVICI

Universitatea “Politehnica” din București

Mobile devices are being used increasively these days. More and more software is being ported to these devices. Everyone knows that creating one’s own radio station is just a few clicks away, using the computer and Internet. But what if we could create a radio station stat could fit into our pockets? Imagine a portable radio station. Incredible?! No quite, just a pocket pc away. Using the newest technologies provided for pocket pc software development, we have built a fully functional steaming server. The paper will explain how it is built, and how it can be used. Moreover, there are several difficulties one has to handle when writing such a server. These will be discussed also.

I. Introducere

Dispozitivele mobile devin din ce în ce mai folosite. Dacă acum câțiva ani, lumea putea doar visa la idea de a asculta muzică pe telefon, acum există deja dispozitive mai mici decât telefoanele care pot reda filme. Lumea este plină de dispozitive *inteligente*.

Este foarte probabil că dispozitivele mai sus amintite, mici și foarte portabile, vor înlocui încet încet celebrele calculatoare personale. Deja un calculator portabil este considerat *mare*. Mai mult, arhitectura lor (ARM) permite o durată de funcționare autonomă mult mai mare, iar sistemele de operare pentru ele (Symbian, Windows CE etc.) sunt foarte puternice.

Odată cu evoluția sistemelor mobile, au evoluat posibilitățile și limbajele de programare pentru acestea. Astfel astăzi avem la dispoziție limbaje ca C/C++, C# sau Java pentru dispozitive inteligente.

Proiectul nostru își propune studierea posibilității creării unui server radio pentru dispozitive portabile. Am implementat server-ul folosind două limbaje diferite, C# și C/C++, și am comparat performanțele obținute.

II. Server

Ce înseamnă un server radio? Odată cu dezvoltarea rețelei Internet și a protocolului HTTP, funcțiile oferite de calculatoare au crescut exponențial. Astfel, astăzi se se pot citi cărți, vedea filme, emisiuni TV (înregistrate sau în direct), asculta posturi de radio folosind un calculator legat la Internet.

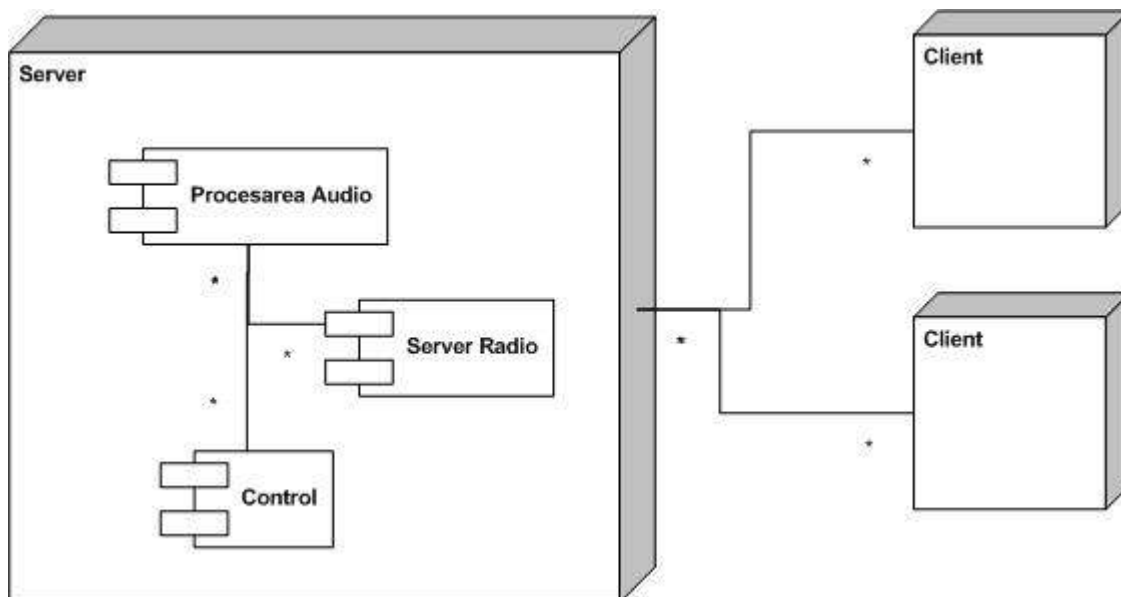
Pentru ca posturile de radio să poate “emite” pe Internet este nevoie de un program special, care are rolul de emițător. Poate fi făcută o analogie cu o stație de transmise de unde radio. Acesta este server-ul radio.

Există mai multe protocoale disponibile pentru emisie pe Internet: Shoutcast, Icecast, RTP etc. Programul nostru implementează protocoalele HTTP, Shoutcast și un protocol creat special de către noi pentru rețele locale.

Informația audio este formată din fișiere mp3. Deocamdată server-ul nu suportă preluare de informație audio din alte surse.

III. Structura

Server-ul este format din trei componente: procesarea informației audio ce va fi trimisă pe Internet, server-ul care se ocupa de transmisia propriu-zisă și un server de control, are permite controlarea de la distanță.



Procesarea informației audio

Informația audio este obiectivul principal al server-ului. Ea este obținută din fișiere mp3. Componenta de procesare citește fișierele și stochează informația audio din acestea într-un buffer. Fișierele mp3 conțin date suplimentare, cum ar fi titlul melodiei, autorul, versuri etc. Acestea sunt citite și memorate de asemenea. Până în prezent sunt suportate doar informațiile ID3 Tag V1.0 și ID3 Tag V1.1.

Un sistem de cu feedback realizează sincronizarea vitezei de citire din fișier și de umplere a buffer-ului. Informația trebuie încărcată în buffer cu viteza cu care trebuie redată. Fără acest sistem, citirea unui fișier s-ar realiza instantaneu. De asemenea, sistemul permite astfel citirea și procesarea fișierelor cu rată de biți variabilă (VBR).

Server propriu-zis

Datele audio stocate în buffer sunt transmise programelor de redare prin intermediul componentei de server audio. Protocolul utilizat este HTTP sau Shoutcast (dacă suportă programul de redare de la client, Shoutcast fiind similar cu HTTP).

Server-ul crează câte un fir de execuție pentru fiecare client care dorește să asculte postul de radio.

Spre deosebire de alte programe similare, componenta noastră de server suportă fișiere cu rată de biți variabilă (VBR). Astfel calitatea semnalului audio este mai bună. Apar totuși dificultăți cu unele programe client mai vechi care nu știu de această facilitate.

Controlul

Programul create de noi rulează sub forma unui serviciu de sistem, ne având interfață grafică. Controlul lui se realizează în două feluri: prin pagină web sau printr-un program special.

Prima modalitate presupune implementarea unui mini server web. Astfel programul poate fi controlat din pagină web. Dezavantajul este că o pagină web este statică, nu permite monitorizare în timp real.

Pentru a urmări server-ul în timp real, am creat un protocol special și un program care poate monitoriza server-ul. Protocolul este în mod text, astfel putând fi folosit și prin telnet.

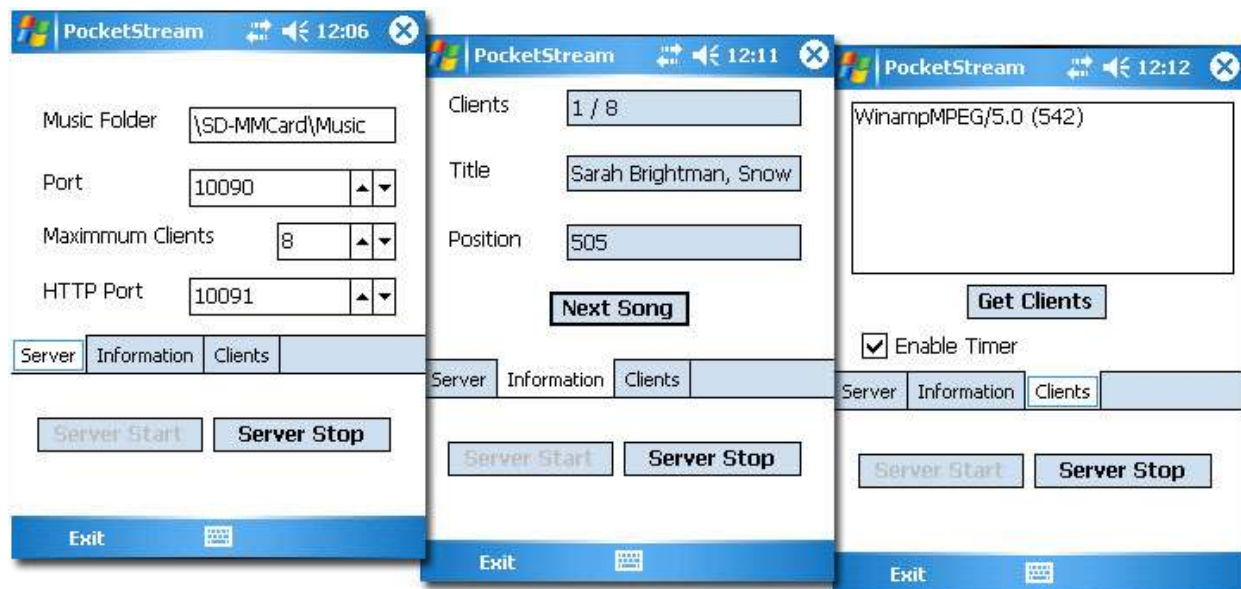
IV. Implementarea

Platform pentru care am creat acest program este Windows CE. Acest sistem de operare suportă limbajele de programare C# cu Compact Framework și C/C++ utilizând biblioteci similare cu cele ale Windows pe 32 de biți.

C Sharp

Primul test l-am realizat folosind C#. Limajul este foarte portabil și are foarte multe pachete puse la dispoziția programatorului. Practic implementarea a fost foarte simplă și rapidă. Programul funcționează astfel și pe calculatoare personale, fără modificări majore (interfața grafică treuie modificată).

Resursele folosite însă de acest limbaj de programare sunt foarte mari. Ajungem până la ocuparea a peste 30 MB de memorie. Am încercat diverse optimizări, refolosiri de obiecte, forțări de dealocare de memorie. Rezultatul a fost aproape același, se poate realiza dar cu costuri mari de memorie.



În privința stabilității, este excelentă. Dispozitivul mobil însă necesită un processor puternic și memorie multă.

Visual C++ for Pocket PC

A doua variantă de implementare a fost C/C++. Limbajul pune la dispoziția programatorului biblioteci similare cu cele ale Visual C++ pentru Windows. Totuși, implementarea a fost mai lentă, timpul necesar pentru testarea și depanarea codului fiind mai mare.

Performanțele sunt însă incomparabile. Memoria ocupată este de aproximativ 1 MB plus dimensiunea buffer-ului audio.

Stabilitatea este bună, apar însă câteva probleme la calcularea sincronizării. Windows CE are rezoluția de ceas de o secundă. De asemenea, generarea numerelor aleatoare lasă de dorit, precum se întâmplă și în Windows pentru calculatoare personale.

În privința portabilității, deoarece Windows CE are biblioteci similare cu cele ale Windows pentru calculatoare personale, programul poate fi compilat cu mici schimbări și pentru acesta. Performanțele față de varianta C# pentru calculatoare personale sunt evidente.

V. Stație radio în rețele locale

Deoarece stațiile de radio care folosesc dispozitive mobile sunt mai ales pentru rețele locale, am testat posibilitatea implementării unui protocol radio cu latență foarte mică. HTTP și Shoucast sunt implementate peste TCP/IP astfel viteza de transmisie a datelor nu este suficient de mare. Există o latență de câteva secunde.

Să presupunem că avem un dispozitiv inteligent pe care există stocată muzică. Dorim să ascultăm acesată muzică pe mai multe difuzoare amplasate în apropierea noastră, dar fără să folosim fire de transmise. Nu putem folosi nici unul din protocalele radio standard, datorită latenței. Difuzoarele ar fi desincronizate.

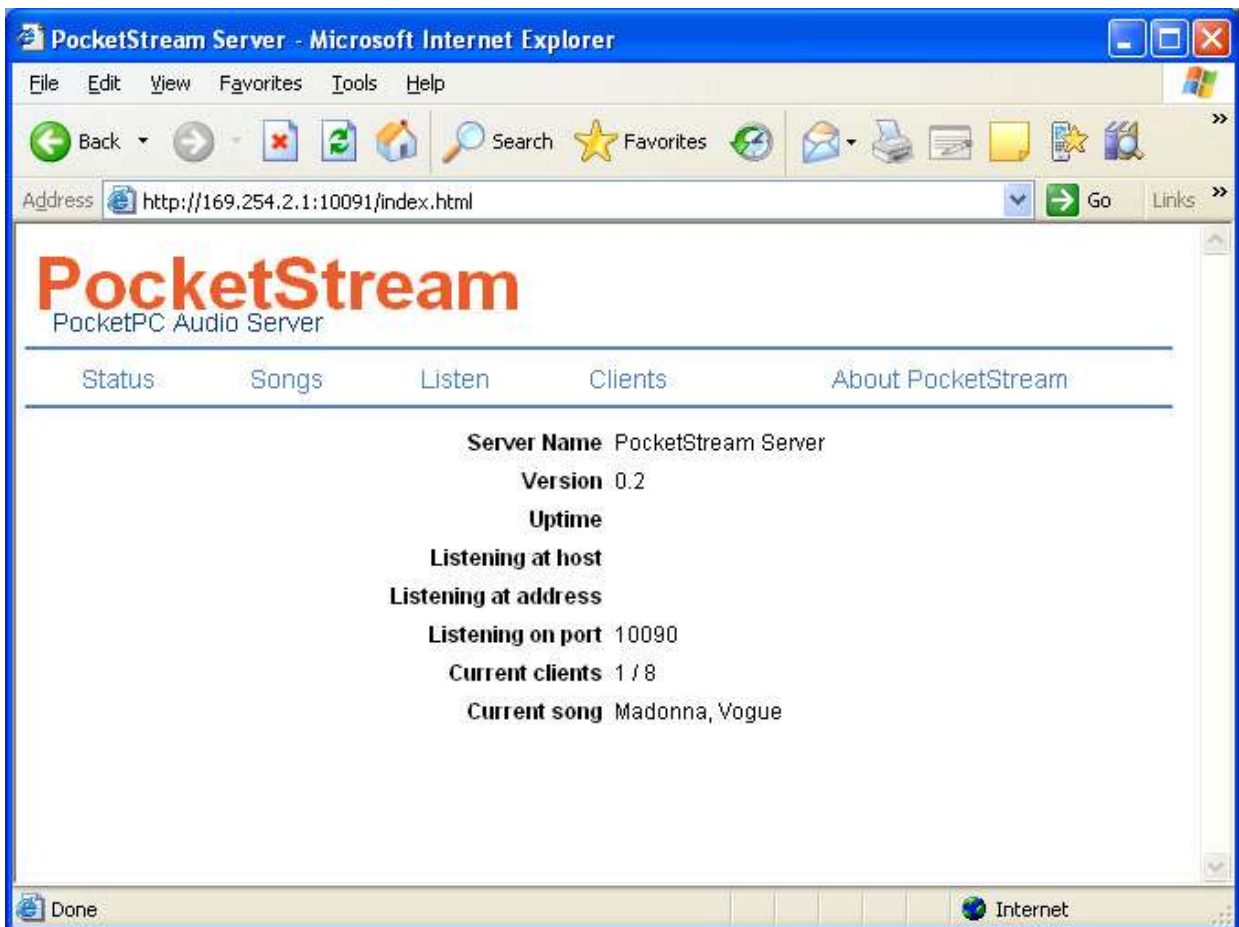
Pentru a rezolva acesată problemă am implementat un protocol care folosește pachete UDP. Aceste pachete sunt transmise fără confirmare de primire, având astfel viteza de trasmisie mult mai mare. Deoarece dorim să le transmitem într-o rețea locală, vom presupune că pierderea de pachete este de maxim 2%, ceea ce este acceptabil.

Folosind acest protocol, server-ul va transmite continuu pe rețeaua locală pachete cu date audio tuturor calculatoarelor. Pentru a asculta stația de radio, utilizatorul trebuie să dețină un porgram care va recepta acesa pachete. Astfel se realizează efectiv o transmise radio similar cu o antenă clasică.

Protocolul funcționează rezonabil în rețele locale, apărând puține erori de transmitere. În rețele mai mari însă, rețele metropolitane sau de întindere mare, protocolul nu este utilizabil.

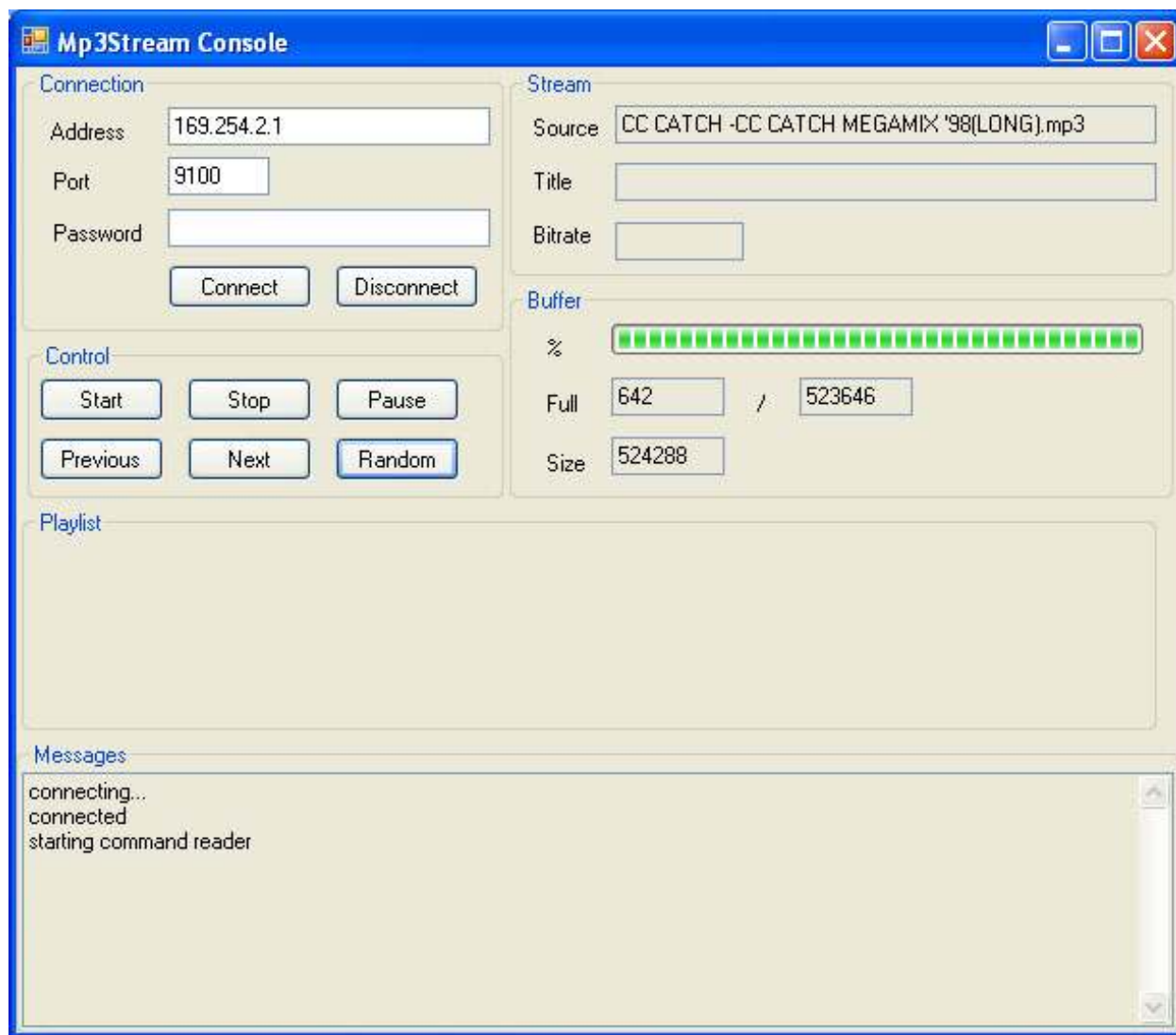
V. Controlul

Am realizat două programe de control, unul în pagină de internet și unul ca program normal.



Pagina de Internet are avantajul că poate fi accesată de pe orice calculator, care rulează orice sistem. Dezavantajul este modul de funcționare al protocolului HTTP. Nu permite monitorizare în timp real.

Programul pentru calculator permite monitorizarea în timp real. Problema este dependența de platformă. El funcționează doar sub Windows. Am realizat și o varietate pentru Pocket PC.



VI. Obiectivul

Obiectivul creării acestui server a fost testarea posibilității de implementare a unor server-e pe dispozitive mobile. Este puțin probabil ca o stație de radio să folosească un dispozitiv mobil pentru a emite. Gândiți-vă însă la o stație de radio pentru un campus. Este nevoie de o rețea locală și un simplu dispozitiv inteligent.

O posibilitate de extindere a acestui program este retransmiterea stațiilor de radio de pe Internet. Astfel devine posibilă retransmiterea într-o rețea locală a unei stații de radio prin simpla folosire a unui simplu dispozitiv.

VII. Concluzii

În concluzie putem afirma că experimentul nostrum este reușit, am demonstrat că este posibilă crearea unui server radio pentru dispozitive mobile. Mai mult, din testul nostru rezultă

că folosirea limbajelor de similitudine cu C# sau Java este încă problematică, iar C/C++ rămâne încă limbajul cel mai potrivit.

Dispozitivele mobile inteligente vor fi din ce în ce mai folosite. Posibilitatea creării unor astfel de programe pentru ele demnstrază utilitatea lor, și întăresc ideea că vor concura serios calculatoarele personale.

VIII: Bibliografie

1. YAO, Paul, DURANT, David, *.NET Compact Framework Programming with C#*, Prentice Hall PTR, 2004
2. ***, http://www.mpgedit.org/mpgedit/mpeg_format/mpeghdr.htm
3. ***, <http://www.msdn.com>
4. ***, <http://www.pocketpcdn.com>
5. ***, <http://www.codeproject.com>
6. ***, <http://www.shoutcast.com>
7. ***, <http://www.nullsoft.com>