

## APLICAȚII OMNIPREZENTE ÎN AMILAB

Student: Claudiu-Mihai TOMA<sup>1</sup>

Conducător științific: prof.dr.ing. Andrei OLARU<sup>2</sup>

*O problemă comună în zilele noastre o reprezintă dificultatea de a transmite date de la dispozitivele mobile către calculatoare. Sunt necesare cabluri, configurare manuală, drivere, utilizarea de servicii centralizate ce folosesc conexiunea la Internet sau alte inconveniente ce afectează experiența utilizatorului.*

*O soluție simplă, ușor de utilizat și intuitivă o reprezintă o aplicație distribuită care realizează automat o conexiune între un dispozitiv mobil și o stație prin simpla direcționare a camerei telefonului către calculatorul destinație, după care se începe automat transmisia datelor, utilizatorul având astfel o intervenție minimală.*

**Cuvinte cheie:** transfer de date, automatism, ușurință în utilizare, calculator, telefon inteligent

### 1. Introducere

Aflându-ne în epoca tehnologiei, dorim ca dispozitivele noastre să comunice cu cât mai mare ușurință, un caz particular fiind transmisia de date precum link-uri sau imagini, de la telefoanele inteligente către stațiile de tip desktop sau laptop.

Această lucrare tratează problema menționată anterior, având ca scop obținerea unei aplicații distribuite ce va permite utilizatorilor să transmită informații prin simpla direcționare a camerei telefonului către stația destinație.

În prezent, aplicația poate transmite conținut sub formă de șiruri de caractere către stații care se pot afla în următoarele două situații:

a) stații al căror ecran este vizibil, fiind afișat un cod QR pentru identificare;

b) stații al căror ecran nu este vizibil dar care dispun de o cameră web ce va furniza imagini folosite de aplicație pentru a identifica în mod unic fiecare calculator.

Vom numi această aplicație „Sherry”.

---

<sup>1</sup> Anul 4, grupa 343C1, Facultatea de Automatică și Calculatoare, email: tomaclaudiu@protonmail.com

<sup>2</sup> Departamentul Calculatoare, Facultatea de Automatică și Calculatoare, e-mail: cs@andreiolaru.ro

Spre deosebire de majoritatea programelor de pe piață care necesită a fi configurate sau au nevoie de cabluri, aici conexiunile între dispozitive se realizează automat, dinamic, fără a necesita intervenția utilizatorilor. Singura condiție este aceea ca toate dispozitivele să se afle în aceeași rețea locală.

Atât datorită constrângerilor cauzate de diferența mare între platforme cât și de funcționalitatea foarte diferită pe care o pun la dispoziție cele două categorii de dispozitive, vom implementa practic două aplicații, una pentru telefoane și una pentru calculatoare, încercând pe cât posibil să păstrăm o simetrie a codului sursă.

Pentru a putea înțelege mai ușor soluția propusă în această lucrare, vom evidenția mai întâi toate punctele cheie, urmând ca acestea să fie explicate în detaliu în capitolele următoare.

În primul rând vom menționa tehnologiile utilizate în dezvoltarea aplicației. Urmează să descriem câteva particularități ale aplicației Android și al celei pentru calculator, modul de formare al topologiei și nu în ultimul rând vom vorbi despre cele două moduri curente de funcționare ale aplicației: folosirea de coduri QR și folosirea de imagini de la camera web pentru identificarea destinației.

## **2. Descrierea componentelor**

### **2.1 Tehnologii**

Având în minte faptul că Sherry trebuie să fie o aplicație ușor de utilizat, este obligatoriu să ținem cont de un aspect foarte important: portabilitatea. Astfel, utilizatorul nu își va pune problema compatibilității aplicației cu dispozitivele sale. Tehnologiile au fost alese în acest spirit.

Pe de o parte avem Android SDK API 16, pentru a acoperi peste 98% din dispozitivele cu sistemul de operare Android, dispozitive care reprezintă peste 80% din piața telefoanelor mobile din ziua de astăzi, piață în continuă creștere. Un alt motiv pentru care a fost aleasă platforma Android îl reprezintă comunitatea uriașă de dezvoltatori și ajutorul neprețuit al oamenilor cu experiență oferit dezvoltatorilor începători care doresc să își concretizeze ideile și să le pună la dispoziție consumatorilor.

De asemenea, un limbaj des folosit pentru aplicații portabile este Java, unul dintre limbajele care ajută la ruperea barierei create artificial de diversitatea platformelor. Este folosit pentru aplicația ce rulează pe calculatoare, incluzând biblioteci opensource Java precum webcam-capture de la SarXos sau ZXing (pentru scanare de coduri QR).

O altă bibliotecă utilizată este OpenCV, o bibliotecă puternică, axată pe prelucrarea de imagini având performanțe excelente și o versatilitate crescută.

## 2.2 Aplicația Android

Aplicația de pe telefon anunță sistemul de operare Android faptul că este capabilă să distribuie text. Astfel, utilizatorul poate alege această aplicație pentru a distribui textul respectiv către una din stațiile din apropiere.

Odată pornită, aplicația intră în mod automat în modul de detecție al calculatorului destinație cu ajutorul camerei web.

În momentul de față, pentru a intra în modul de scanare a codurilor QR este necesară apăsarea unui buton. De asemenea, cele două moduri nu pot funcționa concomitent, însă, în viitorul apropiat, acest impediment va fi înlăturat.

Aplicația se oprește după ce datele au fost transferate cu succes.

## 2.3 Aplicația pentru PC

Aplicația de pe calculator așteaptă permanent să primească date pe care le scrie la ieșirea standard. O altă acțiune permanentă pe care o face este să afișeze cadre primite de la camera web. Această funcționalitate din urmă este implementată pentru a ușura depanarea, versiunea finală urmând să capteze cadre (fără a le mai afișa) doar atunci când un telefon anunță faptul că are date de trimis.

Când un telefon intră în modul de scanare QR, toate calculatoarele conștiente de eveniment vor afișa în colțul ecranului un cod QR ce reprezintă propria adresă IP, adresă ce va fi folosită de telefon pentru a-i trimite datele.

## 3. Detalii de implementare

### 3.1 Formarea topologiei

În momentul în care o instanță de Sherry este creată pe un telefon, aceasta trimite periodic mesaje broadcast UDP cu scopul de a-și face cunoscută adresa IP.

Este mult mai eficient ca telefonul să emită aceste pachete deoarece, spre deosebire de calculator, dispozitivul mobil va avea aplicația Sherry pornită doar pe durata identificării destinației și a realizării transferului de conținut. Astfel, rețeaua locală va fi populată un timp minim cu aceste mesaje de broadcast.

Calculatoarele ce vor recepționa astfel de mesaje vor încerca să inițieze o conexiune TCP către adresa IP identificată în pachetul UDP menționat anterior, la un port prestabilit.

Administrarea rețelei are loc la executarea operațiilor de trimitere și recepționare de date. Când una din aceste operații aruncă o excepție, conexiunea asociată este eliminată din lista de conexiuni active.

### 3.2 Mod scanare QR

Primul mod implementat de identificare a calculatorului destinație a presupus utilizarea codurilor QR.

Dacă un utilizator va alege această aplicație când decide să distribuie conținut și alege ca mod de transfer modul QR (activat din interfața grafică prin apăsarea unui buton) vor începe să fie transmise pachete periodice UDP care semnalează calculatoarelor starea în care se află aplicația de telefon.

Toate stațiile care vor recepționa mesajele UDP vor afișa în colțul ecranului un cod QR care encodează propria sa adresa IP din rețeaua locală.

În momentul în care telefonul a scanat cu succes un cod, va folosi rezultatul pe post de adresă IP destinație și va încerca să trimită conținutul prin intermediul unei conexiuni TCP.

Stațiile așteaptă în permanență conexiuni pentru recepționarea de date. Orice se va recepționa va fi afișat la ieșirea standard sub formă de șir de caractere.

Când aplicația de telefon trimite cu succes conținutul la un calculator, aceasta își termină execuția.

### 3.3 Mod comparare imagini

Modul predefinit de identificare a destinației este cel pe baza de comparație între imagini.

Aplicația Android captează în permanență cadre de la camera din spate a telefonului, le afișează utilizatorului și le trimite tuturor calculatoarelor ce se află în topologia descrisă anterior.

Calculatorul, după ce va recepționa un astfel de cadru, va solicita o captură de la propria camera web și va compara cele două imagini. Rezultatul comparației este numărul de puncte cheie comune identificate de OpenCV, unde un punct cheie reprezintă o zonă a imaginii pe care OpenCV este relativ sigur că o va identifica chiar dacă imaginea a fost rotită, a fost făcută de la o distanța mai mare sau de la un unghi ușor diferit. Acest număr este trimis telefonului.

Telefonul colectează toate aceste rezultate și alege calculatorul care a identificat cele mai multe similarități. Acest calculator va fi destinatarul conținutului.

## 4. Concluzii

Cu toate că Sherry are deficiențe în stadiul actual, cum ar fi faptul că utilizatorul va putea vedea în jur de două cadre pe secundă, am constatat faptul că aplicația este foarte ușor de utilizat datorită faptului că acțiunile făcute de utilizator sunt intuitive. Pe lângă asta, detecția codurilor QR funcționează extrem de bine iar modul de identificare pe bază de imagini are rezultate promițătoare.

În general s-a optat pentru soluții simple, ușor de depanat și extins. Spre exemplu, chiar dacă în prezent se transmit doar șiruri de caractere, se poate adăuga foarte ușor posibilitatea de a transfera orice tip de fișier prin encodarea conținutului său în Base64.

Doarece acest proiect reprezintă o lucrare de licență în desfășurare, există numeroase lucruri care vor fi îmbunătățite și cercetare ce va fi efectuată pentru a extinde funcționalitatea și calitatea soluției finale.

Printre funcționalitățile ce vor fi implementate se numără:

1. posibilitatea transferării de conținut de la calculator către telefon;
2. modul bazat pe comarația imaginilor și modul QR vor funcționa în paralel;
3. se va putea transfera orice tip de fișier;
4. vor fi evidențiate calculatoarele pe ecranul telefonului.

## BIBLIOGRAFIE

- [1]. *Android Developers*. 31 Mar. 2017. <https://developer.android.com/index.html>. 9 Apr. 2017
- [2]. „Java™ Platform, Standard Edition 7. API Specification.” *Overview (Java Platform SE 7)*. 13 Apr. 2017. <https://docs.oracle.com/javase/7/docs/api/overview-summary.html>. 9 Apr. 2017
- [3]. *OpenCV: OpenCV Tutorials*. 11 Mai 2017. [http://docs.opencv.org/master/d9/df8/tutorial\\_root.html](http://docs.opencv.org/master/d9/df8/tutorial_root.html). 6 Mai 2017.
- [4]. Firyn, Bartosz. *Webpage Capture in Java*. 21 Ian. 2017 <http://webcam-capture.sarxos.pl/>. 9 Mai. 2017
- [5]. *All Classes (Zxing 3.3.0 API)*. 5 Mai 2017. <https://zxing.github.io/zxing/apidocs/allclasses-iframe.html>. 3 Mai 2017
- [6]. Mahmoud, Qusay H. „Sockets programming in Java: A tutorial.” *Sockets programming in Java: A tutorial | JavaWorld*. 11 Mai 2017. <http://www.javaworld.com/article/2077322/core-java/core-java-sockets-programming-in-java-a-tutorial.html>. 16 Apr. 2017