

Monitorizarea traficului

Bogdan Zavadovschi

Facultatea de Informatică,
Universitatea "Alexandru Ioan Cuza" Iași,
bogdan.zavadovschi@info.uaic.ro

Abstract

Monitorizarea traficului este un proiect care consta in implementarea unei aplicații prin care un participant la trafic poate transmite si primi informații despre diverse evenimente rutiere, transmitând totodata viteza și locația la un interval regulat de timp. Totodată acesta poate sa primească și diverse informații despre vreme, prețul combustibilului și ultimele evenimente sportive. Aceasta este alcătuită din două componente: un client și un server. Ambele folosesc tehnici de concurență, serverul acceptând mai multe conexiuni simultan, iar clientul poate transmite și primi date în același timp.

1 Introducere

Aplicația de monitorizare a traficului constă în două componente cheie: un client și un server, ambele implementate concurent. Clientul are capacitatea de a procesa mesajele venite de la server și de a transmite informații în mod simultan, fără a fi nevoie să se aștepte terminarea unei cereri. Totodată acesta **simulează** doi sensori: unul de viteză și unul de locație. Acești senzori vor fi interogați la un interval fix de timp pe care fiecare client îl va putea seta, în funcție de preferință, iar informațiile vor fi transmise către server. Pe baza informațiilor primite serverul va returna incidentele din trafic (dacă există în regiunea sa) și va modifica ultima locație a utilizatorului în baza de date.

Serverul are posibilitatea de a suporta mai mulți clienți simultan, prin implementarea concurență reușind să răspundă la mai mulți clienți în același timp. La un interval regulat de timp, acesta va trimite informații către toți clienții (care au selectată opțiunea) informații despre vreme, evenimente sportive și ultimele prețuri la combustibil. În cazul în care este raportat un incident, serverul va transmite tuturor utilizatorilor din regiunea specificată evenimentul.

Având imaginea de ansamblu, detaliile privind implementarea și arhitectura aplicației vor fi prezentate în următoarele secțiuni, totodată fiind prezentate și anumite îmbunătățiri pe partea de logică și performanță a aplicației.

2 Tehnologii utilizate

2.1 Protocolul de comunicare

Comunicarea dintre client/clienti și server este realizată prin **Transmission Control Protocol (TCP)**. Decizia de utilizare a acestui protocol provine din faptul că toate pachetele de informație trebuie să ajungă în aceeași ordine, ceea ce este important întrucât pe baza datelor primite de server este implementată logica de transmitere a evenimentelor rutiere, iar mesajele de tip text primite de client trebuie să fie inteligibile.

2.2 Stocare datelor

Pentru a putea ține o anumită evidență a datelor, atât a utilizatorilor, cât și a evenimentelor media, incidente rutiere, limite de viteză șamd. aplicația utilizează o bază de date, *sqlite3* ce asigură persistența și stocarea datelor. Folosindu-se de limbajul SQL, abstractizările la nivelul implementării sunt relativ ușor de făcut, scurtând timpul de implementare.

2.3 Securitate

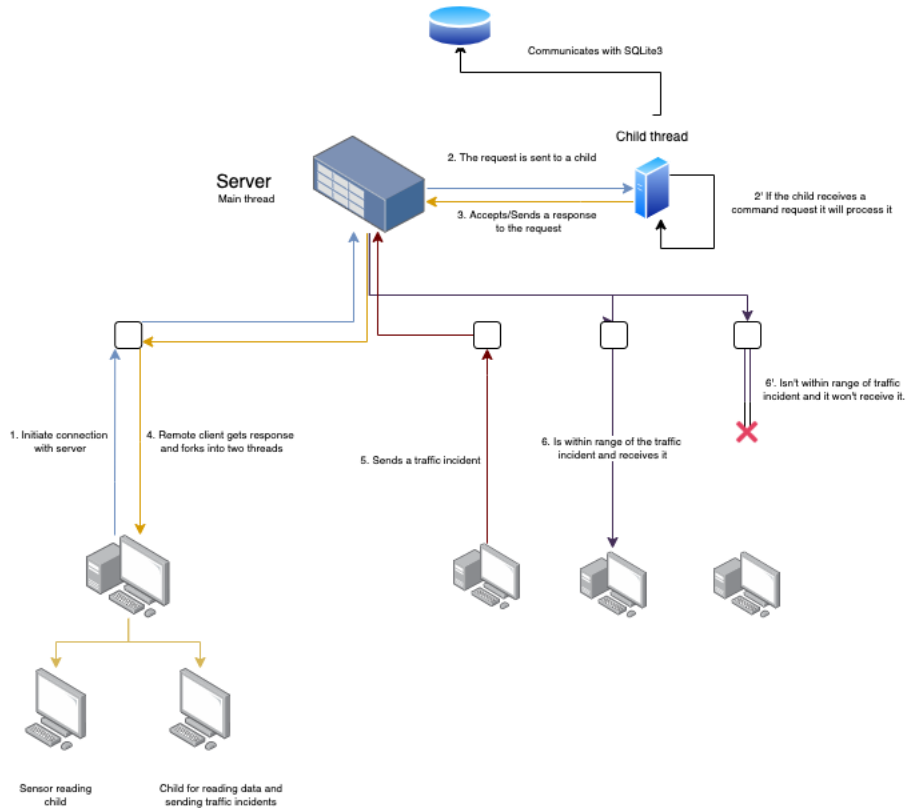
Din motive de securitate, odată ce avem conexiunea între client și server, vom crea un nou context de tipul SSL, transmitând astfel datele criptate. Pentru acesta vom folosi librăria *OpenSSL* din C++. Totodată stocarea parolele în plain-text nu este o soluție bună pentru zilele noastre, astfel că din aceeași librărie vom folosi algoritmul de hashing **SHA-256**.

2.4 Simularea Senzorilor

În încercarea de a face cât mai ușoară tranziția de la o aplicație de calculator la una de mobile, am adăugat un feature prin care se pot simula doi senzori, unul de viteză și unul de locație, strâns legați între ei. Aceștia pot fi citați, returnând date cu un anumit nivel de corectitudine între ele.

3 Arhitectura aplicației

3.1 Diagrama aplicației



3.2 Arhitectura Clientului

3.3 Arhitectura Serverului

Legat de comunicarea internă a aplicației avem clientul care utilizează două threaduri, unul pentru transmiterea datelor despre viteză și locație (și totodată primirea informațiilor referitoare la limitele de viteză de pe acel segment), iar unul pentru operațiile de bază (raportarea evenimentelor, primirea datelor media) împreună cu Multiplexare I/O pentru comunicarea cu serverul. Pe partea serverului avem Multiplexare I/O pentru comunicarea cu clienții, iar pentru fiecare cerere realizată se va alocă un nou *thread*, urmând să se dea loc după procesarea requestului.

4 Detalii de implementare

5 Concluzii

6 Bibliografie

1. SSL Server-Client using OpenSSL
2. Algorithm for offsetting coordinates by some amount of meters