

Sistem Distribuit de Monitorizare și Asistență a Traficului Rutier în Timp Real

Leonte Alexandru-Valentin

Universitatea „Alexandru Ioan Cuza” din Iași, Facultatea de Informatică

Rezumat Prezenta lucrare descrie proiectarea și implementarea unei aplicații client-server destinate gestionării traficului rutier. Sistemul utilizează protocolul TCP și bibliotecile POSIX Threads pentru a oferi o platformă concurentă capabilă de raportare a incidentelor și monitorizarea telemetriei în timp real.

1 Introducere

Viziunea proiectului constă în crearea unui asistent rutier digital care centralizează datele colectate de la participanții la trafic. Obiectivul principal este facilitarea schimbului de informații în timp real, permițând șoferilor să contribuie activ la o bază de date comună. Sistemul urmărește scalabilitatea prin deservirea simultană a mai multor clienți și asigurarea unui flux de date proactiv, de la server către clienți, prin notificări automate.

2 Tehnologii Aplicate

Implementarea se bazează pe următoarea stivă tehnologică:

- **TCP (SOCK_STREAM):** S-a ales acest protocol datorită necesității de fiabilitate. Mesajele de alertă și datele de autentificare necesită o livrare garantată și ordonată.
- **POSIX Threads:** Serverul utilizează un model de execuție concurent de tip thread-per-client, asigurând un spațiu de adrese comun pentru accesarea rapidă a bazei de date a evenimentelor rutiere.
- **I/O Multiplexing:** Funcția `select()` este utilizată atât pe server pentru gestionarea timpilor de răspuns, cât și pe client pentru monitorizarea simultană a intrării standard și a socket-ului.

3 Structura Aplicației

Arhitectura este modulară, separând logica de comunicare de cea de business. Serverul menține o listă globală de clienți activi, protejată de un Mutex pentru a preveni condițiile de cursă.

Concepte cheie:

- **Worker Threads:** Fiecare conexiune este gestionată de un thread detașat care parsează comenzile clientului.
- **Heartbeat / Monitoring:** Serverul monitorizează actualizările de viteză; dacă un client nu transmite date timp de 70 de secunde, este emis un avertisment automat.
- **Client Speed Thread:** Clientul execută un fir secundar dedicat transmiterii automate a vitezei la fiecare 60 de secunde.

Reprezentarea Fluxului Logic: Conexiunea începe cu faza de autentificare (Faza 1), urmată de actualizarea contextului geografic al utilizatorului (Faza 2). Raportările (Faza 3) declanșează proceduri de tip Broadcast, unde serverul iterează prin lista de sesiuni active pentru a disemina alerta către toți participanții.

4 Protocolul de Comunicare

Protocolul la nivel aplicație este de tip cerere-răspuns, bazat pe text ASCII, cu separatorul „-”. **Comenzi principale:**

- **sign-up-<user>-<pass>:** Înregistrarea unui cont nou.
- **login-<user>-<pass>:** Autentificarea utilizatorului.
- **update-speed-<val>:** Transmiterea vitezei curente.
- **report-<tip>-<strada>:** Semnalarea unui incident.
- **get-info-<categorie>:** Interogarea datelor (vreme, combustibil, sport).
- **subscribe:** Activarea notificărilor asincrone.

5 Scenarii de Utilizare

Scenarii de Succes:

1. **Alertă Viteză:** Clientul trimite **update-speed-85**. Serverul compară valoarea cu limita străzii actuale și returnează un avertisment imediat dacă limita este depășită.
2. **Broadcast Incident:** Un șofer raportează un accident. Toți ceilalți utilizatori primesc mesajul „ALERT: Crash reported!”.
3. **Filtrare Combustibil:** Un utilizator subscris interoghează prețurile pentru o anumită benzinărie sau tip de combustibil, primind datele actualizate direct din **gas.txt**.

Scenarii de Eșec:

1. **Eșec Login:** Utilizatorul încearcă conectarea cu date invalide. Serverul refuză accesul la funcțiile de raportare.
2. **Sesiune Duplicată:** Un utilizator încearcă să se logheze de pe un cont care este deja activ în sistem.
3. **Acces Neautorizat:** Clientul încearcă să obțină informații meteo sau sportive fără a fi utilizat în prealabil comanda **subscribe**.

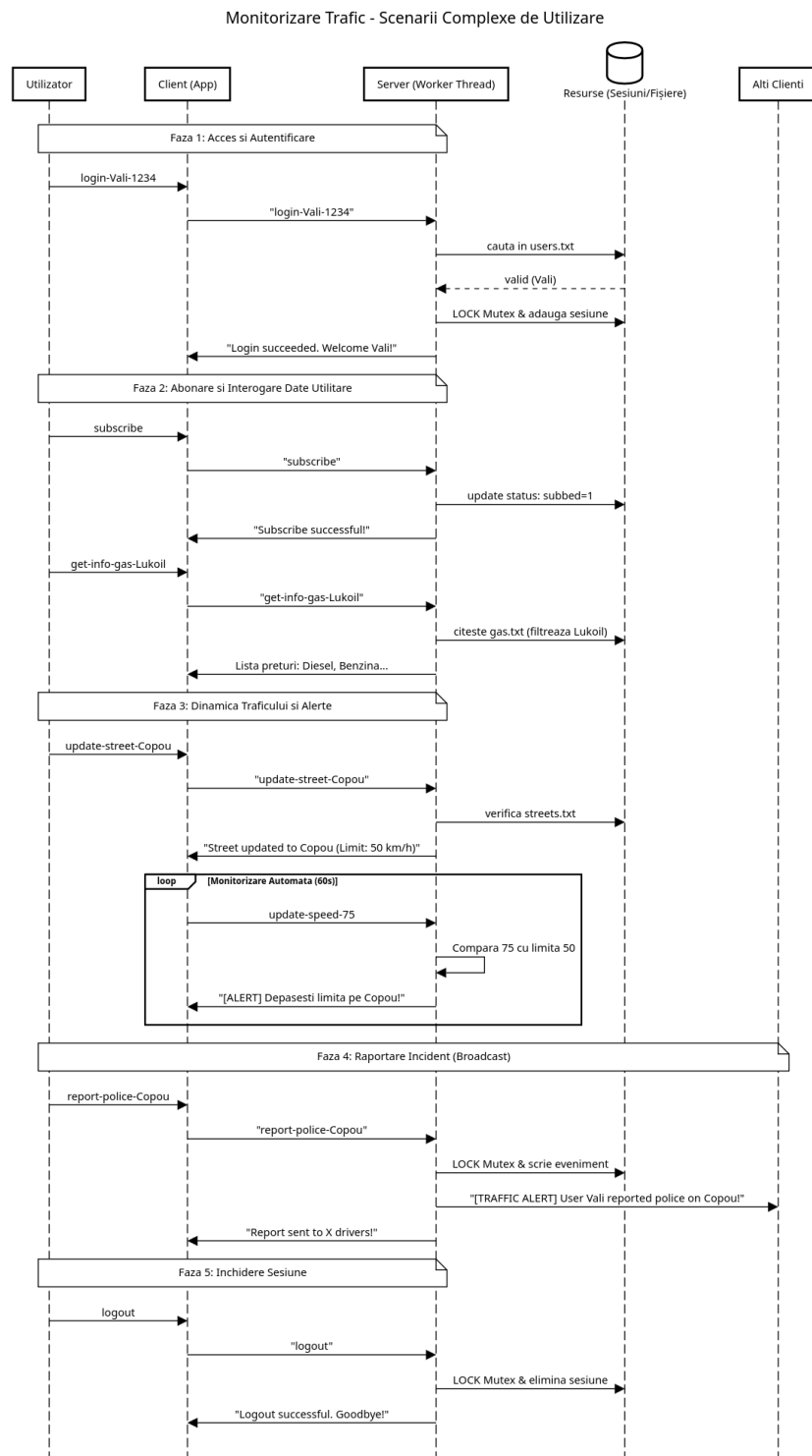


Figura 1. Diagrama de secvență a interacțiunii client-server și fluxul de raportare asincronă.

6 Concluzii și Îmbunătățiri

Sistemul oferă un cadru robust pentru monitorizarea traficului. Ca direcții viitoare, se propune implementarea unui sistem de prioritzare a pachetelor (QoS) pentru a asigura livrarea alertelor de tip „crash” înaintea interogărilor uzuale de tip „get-info” și dezvoltarea unei interfețe grafice (GUI) pentru o vizualizare intuitivă a hărții.

7 Referințe Bibliografice

1. Linux Programmer’s Manual (Man Pages), <https://linux.die.net/man/>
2. Springer LNCS Guidelines, <https://www.overleaf.com/latex/templates/springer-lecture-notes-in-computer-science/kzwwpvhwnvfj>
3. Pagina disciplinei Rețele de Calculatoare, FII UAIC, <https://edu.info.uaic.ro/computer-networks/index.php>.