

Traffic Manager

Student: Mihai Andrei Gherghinescu

Supervizor: Lect. Dr. Todor Ivașcu



Agendă

- Introducere
- Scopuri și direcții viitoare
- Scurt istoric al gestionării traficului
- Abordarea problemei
- Detalii de implementare
- Concluzii și direcții viitoare

Introducere

- **Problemă:** proasta gestionare a traficului
- **Cauză:** numărul ridicat de intersecții în spațiul urban
- **Soluție:** dezvoltarea de sisteme inteligente de gestionare a traficului(ITS)

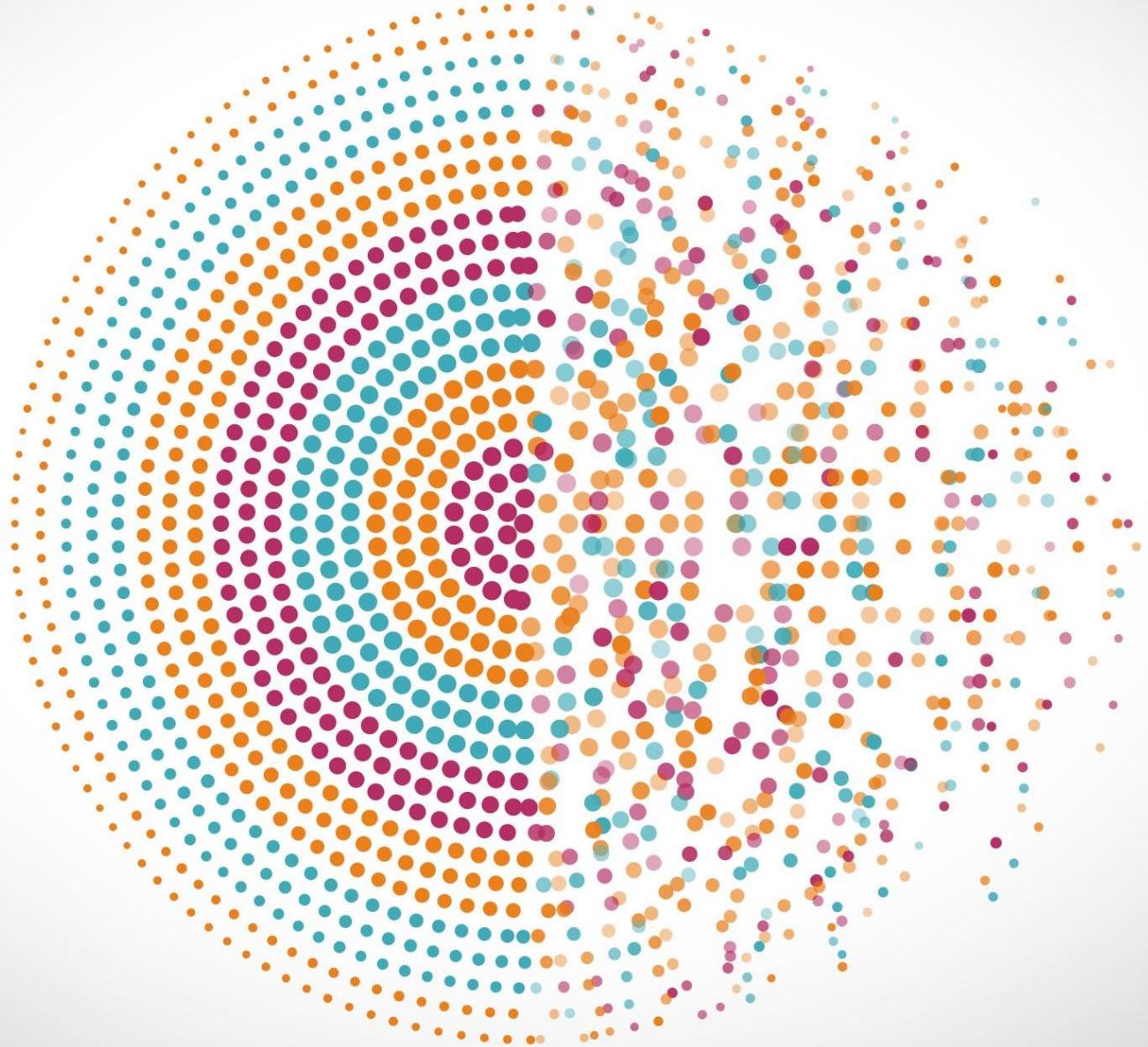


Scopuri și obiective

- Dezvoltarea unui **nou sistem** de semaforizare care să fie:
 - **Performant și accesibil**
 - **Adaptabil la orice condiție de traffic**
 - **Scalabil la nivel global**



Istoric al gestionări traficului

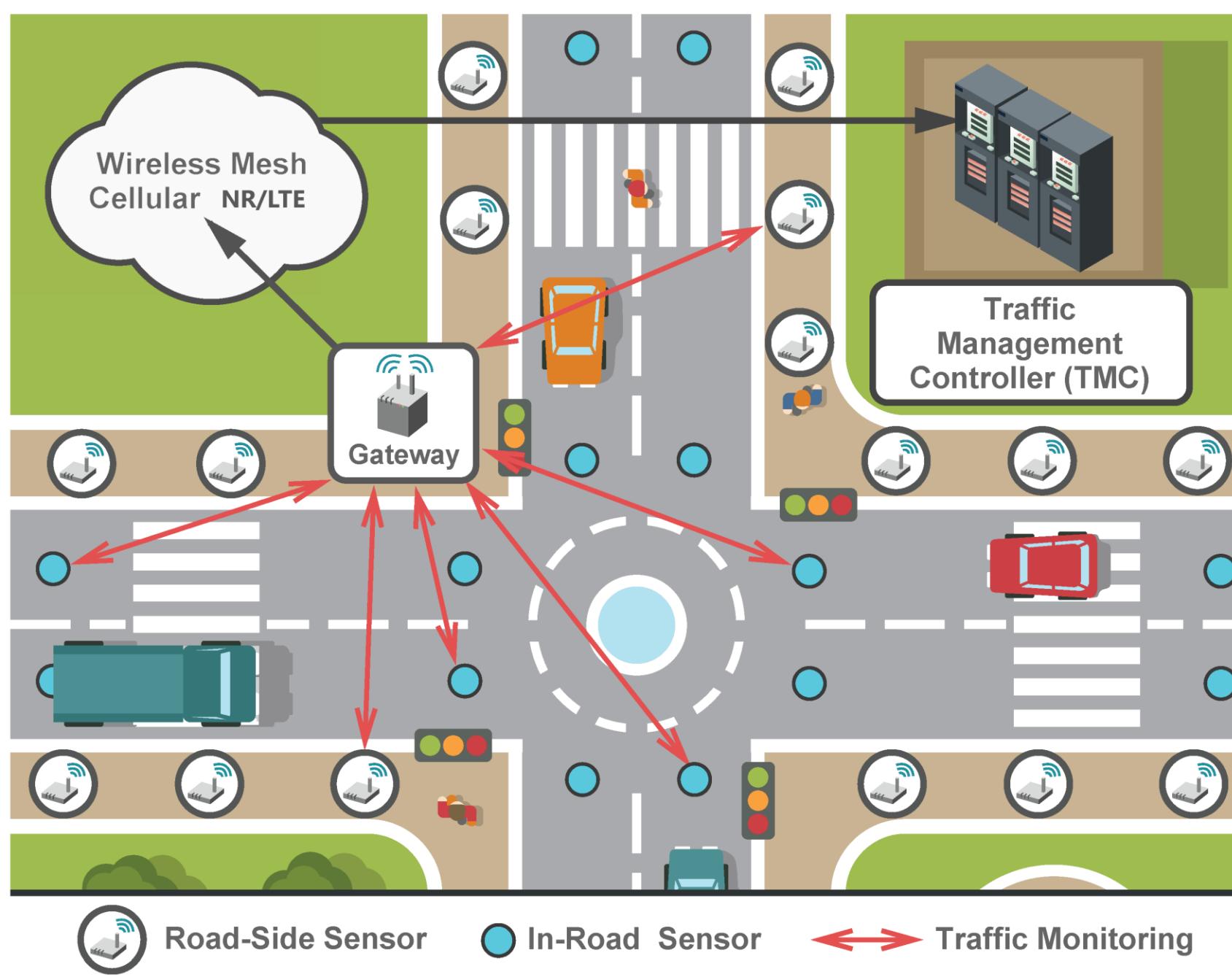


Sisteme bazate pe detecția de obiecte

- Principiu de bază:
 - detectie pe bază de imagini
- Dezavantaje:
 - complexitate computațională mare
 - neeficiente în condiții neprielnice de mediu



Sisteme bazate pe senzori



- **Principiu de bază:**

- plasarea de senzori în carosabil

- **Dezavantaje:**

- costuri mari

Sisteme care sincronizează traficul

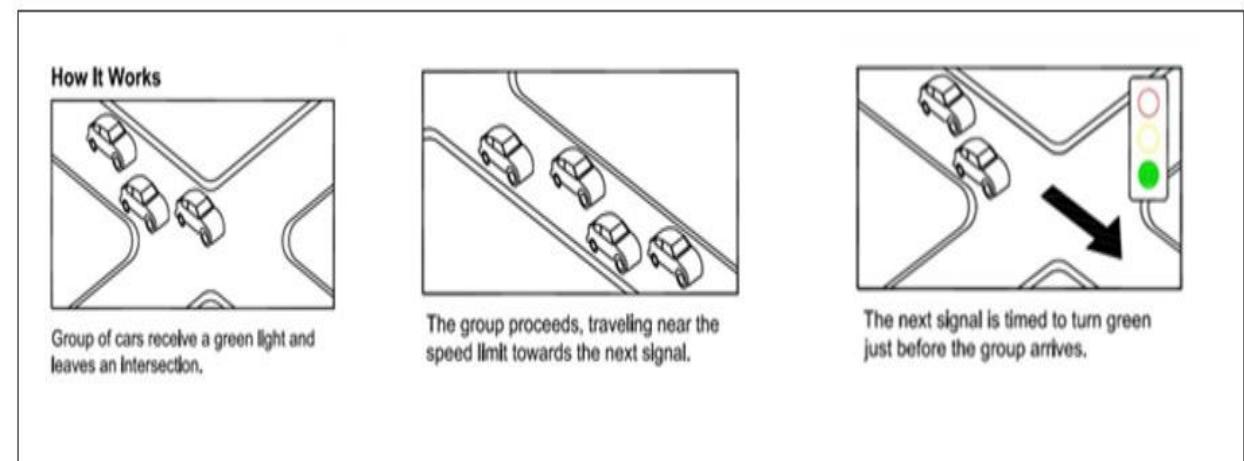
- Principiu de bază:

- număr minim de opriri/porniri

- Dezavantaje:

- presupune viteză constantă a traficului

- ineficiente la intersecția a două rute principale





Sisteme bazate pe logica fuzzy

- **Principiu de bază:**
-aproximarea duratei luminii verzi folosind seturi fuzzy
- Dezavantaje:**
- training real-time de dure mari
 - traficul poate fi îngreunat

Management bazat pe Sisteme de comunicare de acoperire mica(DSRC)

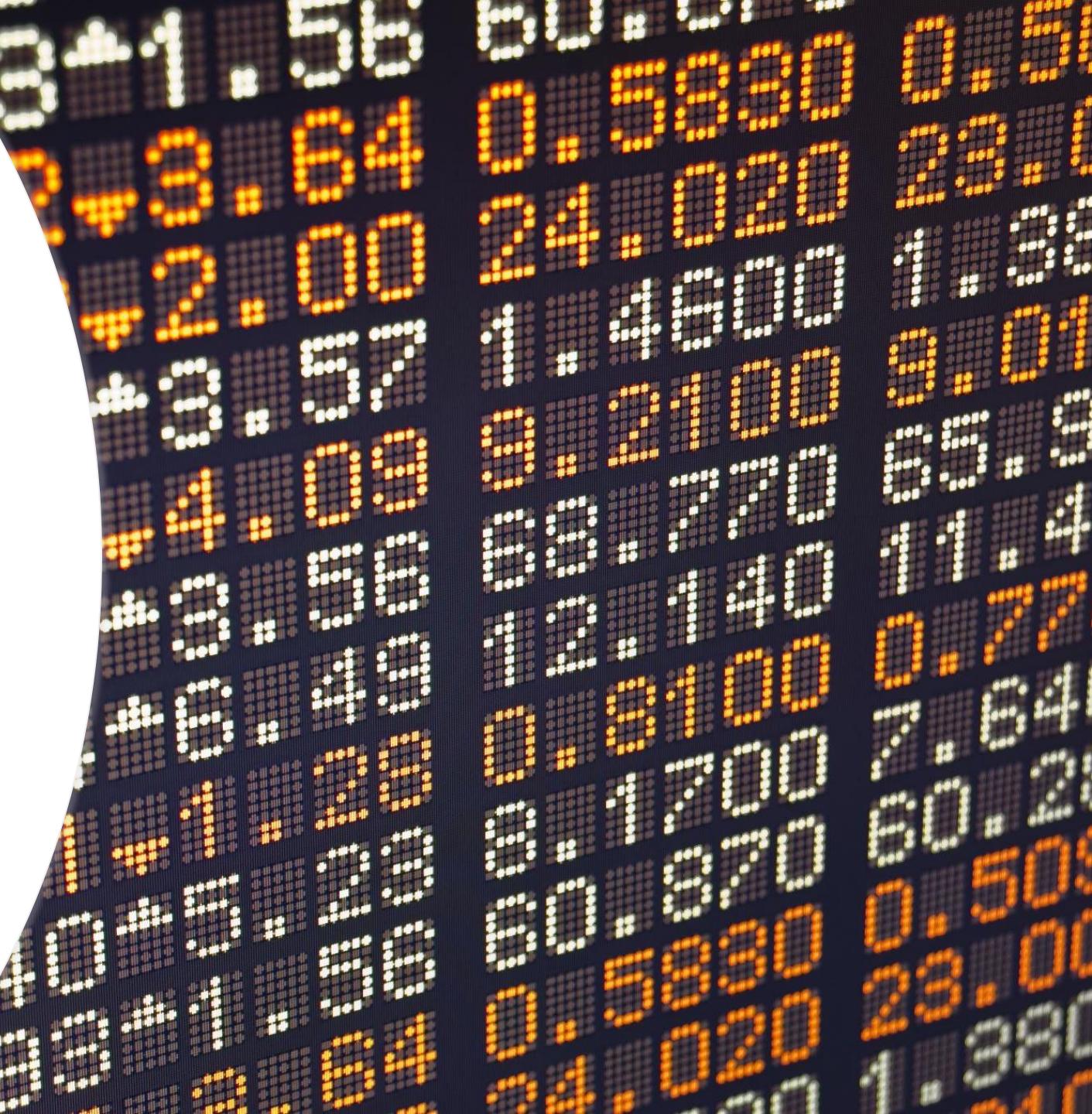
- Principiu de bază:
-folosirea semnalelor radio specializate
- Dezavantaje:
-infrastructura actuală nu permite lansarea acestora

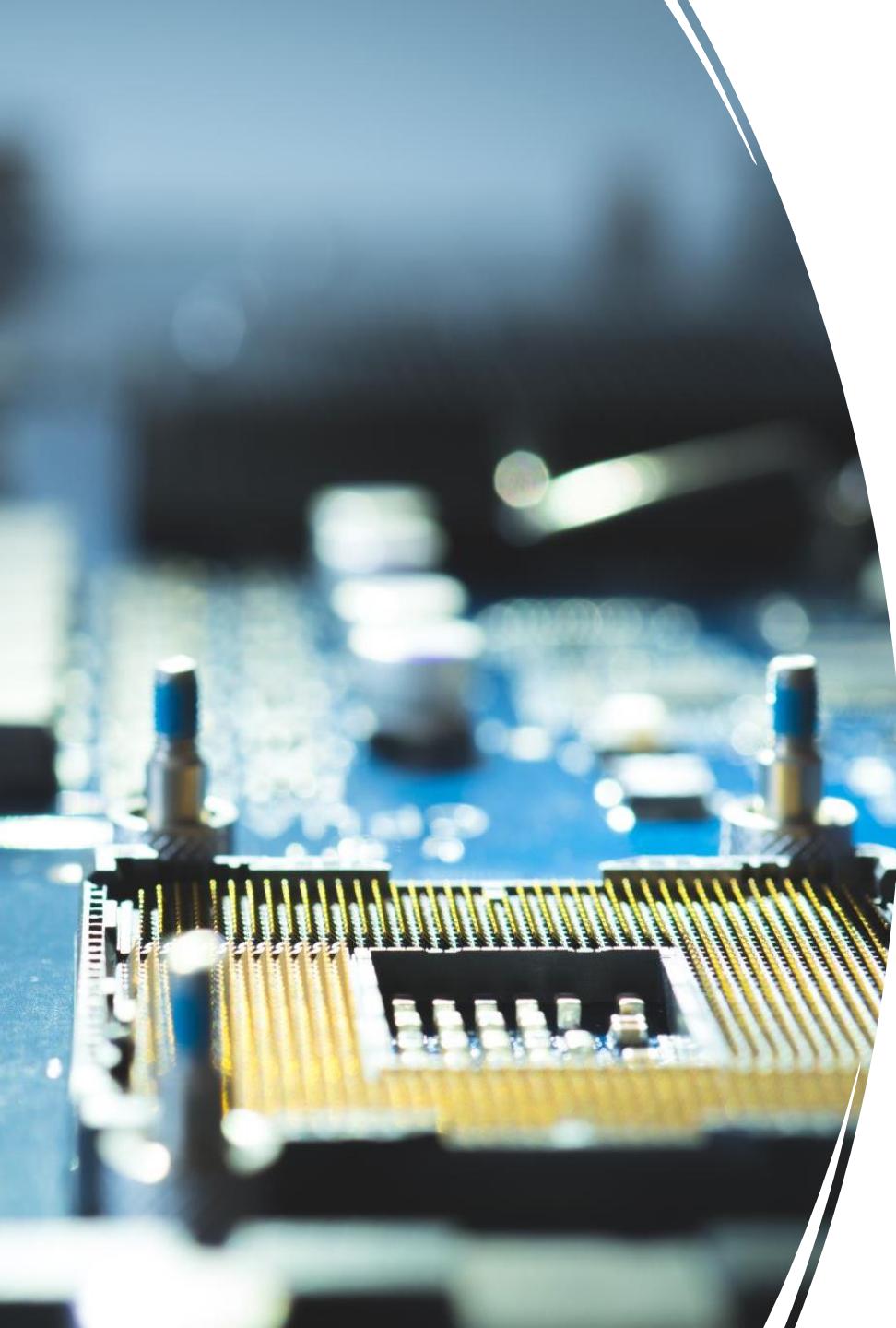


Abordarea problemei

Sistemul nostru de gestionare a traficului

- Extragе punctele forte ale abordărilor prezentate anterior
- Reprezintă o punte infrastructurală către sistemele pe bază de DSRC
- Duce la timpi mai mici de aştepare și poate fi utilizat la nivel global
- Poate funcționa în absența mai multor componente software/hardware



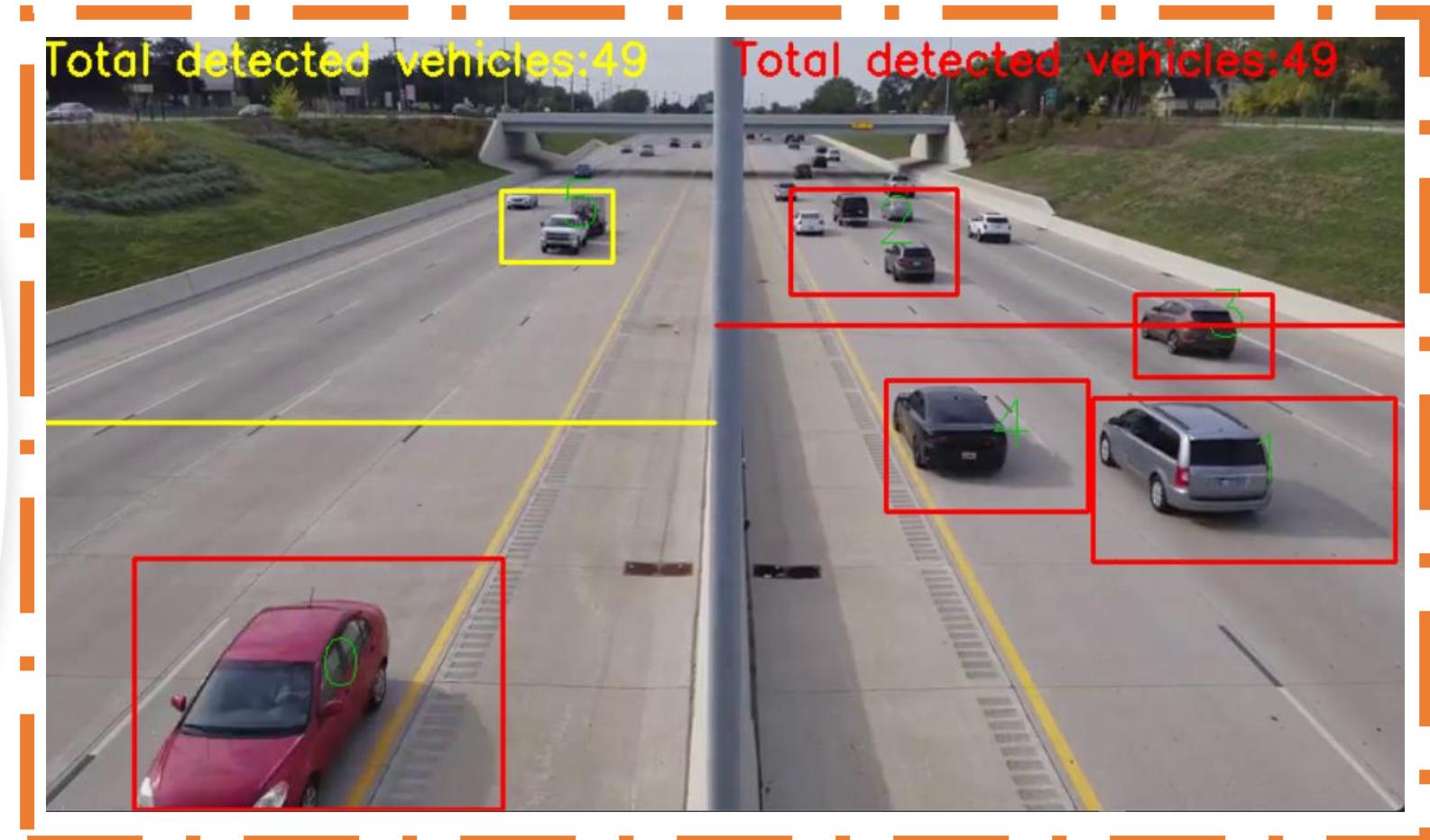


Principii de bază ale sistemului

- Folosește mai multe metode de a **capta date**
- Utilizează **capabilitățile DSRC** ale infrastructurii curente
- Determină **starea traficului** în mod **dinamic**
- **Adaptează durata luminilor** pe baza fluxului de vehicule
- Se bazează pe interacțiune **client-server** pentru a primi datele

Traffic Observer(TO)

- Client conectat la o cameră
- Corespunde unei singure rute de mers
- Colecțează date folosind principii de object-detection
- Poate fi instalat în serie



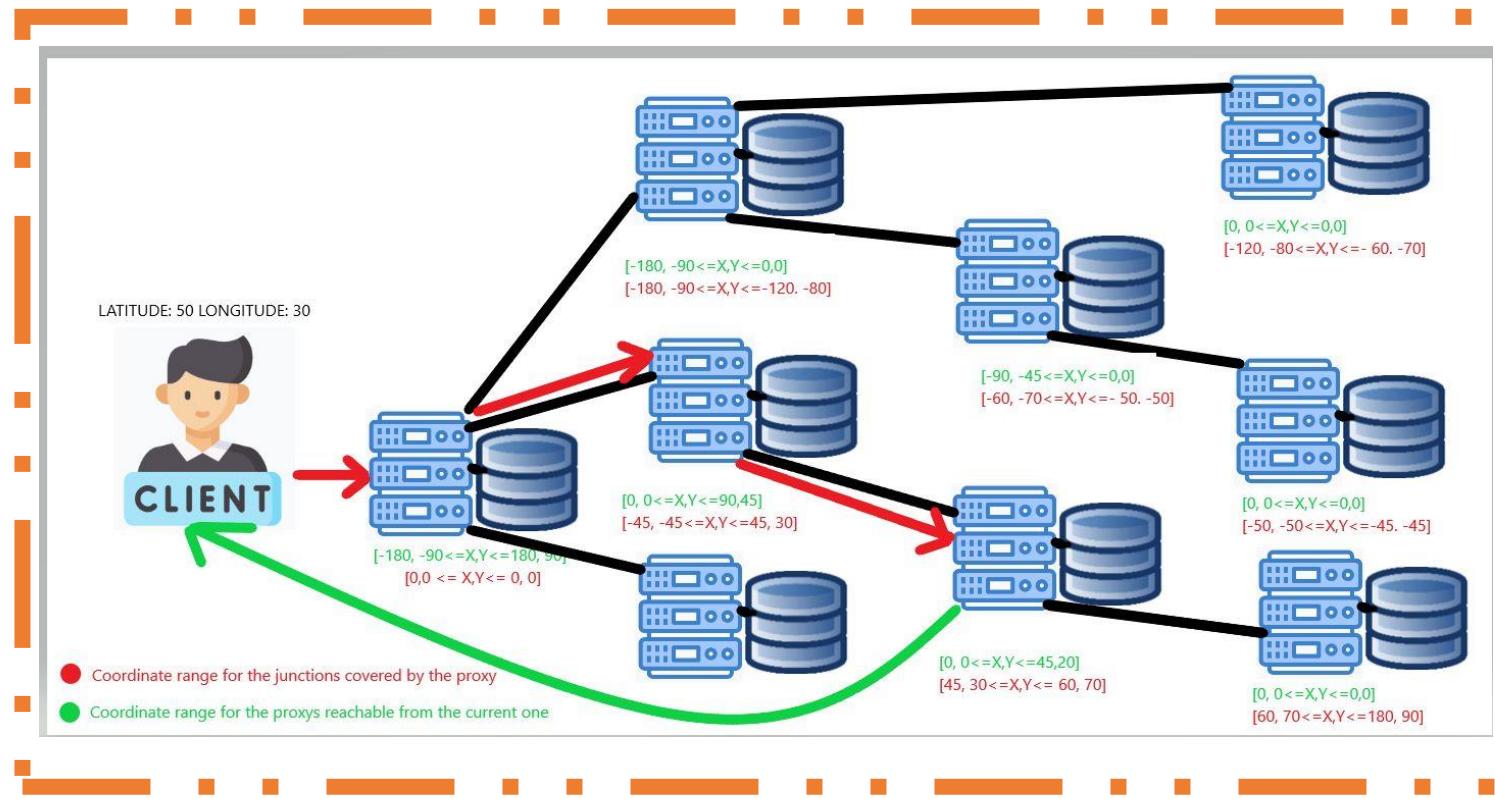
Vehicle Tracker(VT)

- Client instalat direct pe autovehicul
- Folosește date de la GPS pentru a determina poziția cât și direcția de mers
- Același principiu ca și sistemele bazate pe senzori, dar cu costuri reduse
- Determină următoarea intersecție și semnalizează intenția de traversare



Proxy

- Server cu rol de **redirecționare**
- Permite sistemului să fie **scalat pe orizontală**
- Deține o **listă de intersecții și noduri vecine**
- Fiecare proxy poate fi **administrat la nivel regional**
- Minimizează eventualele daune provocate de **atacuri cibernetice**

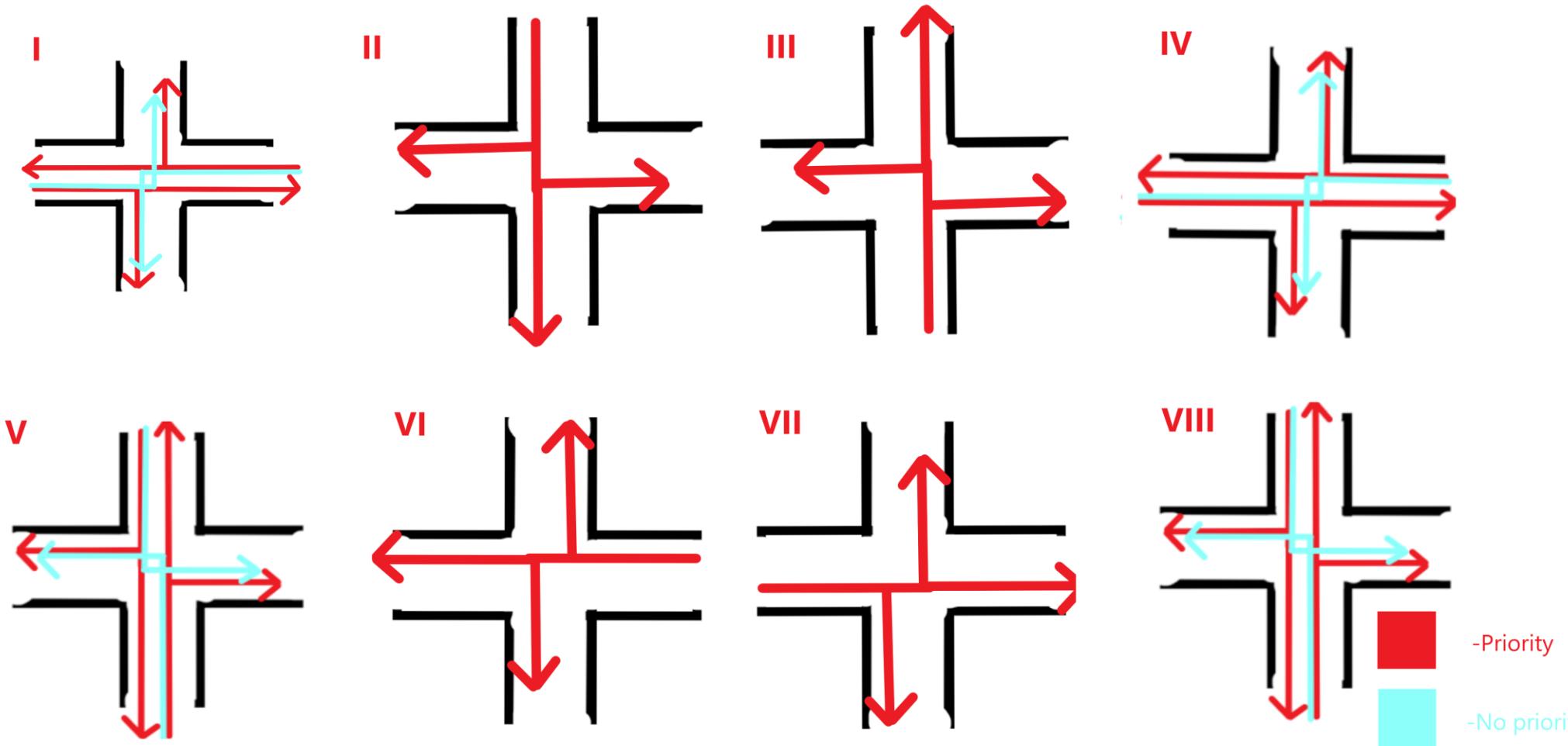


Junction Main Server (JMS)

- Server ce manipulează sistemul de semaforizare
- Instalat la nivel de intersecție
- Captează datele primite de la toți clienți
- Dictează starea traficului
- Adaptează durata semnalelor de verde și de roșu



Stări ale traficului



Modul de alegere al urmatoarei stări

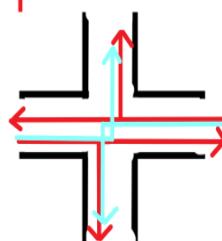
- În **condiții normale**, stările sunt parcuse în mod **ciclic**
- Fiecare direcție de mers are asociată un **cronometru**
- La **finalul luminii verzi** se reevaluatează **starea traficului**
- În caz că unul sau mai multe **cronometre expiră** traficul va “**sări**” la **starea corespunzătoare**.



Exemple de tranzitii ale traficului

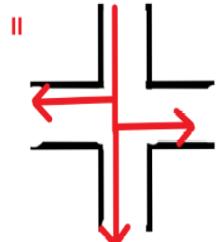
Tranzitie normală

CURRENT STATE:
I
TIMER_E: 120 sec
TIMER_W: 120 sec
TIMER_N: 120 sec
TIMER_S: 120 sec



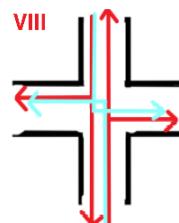
END GREEN
LIGHT PHAZE

CURRENT STATE:
II
TIMER_E: 120 sec
TIMER_W: 120 sec
TIMER_N: 10 sec
TIMER_S: 5 sec



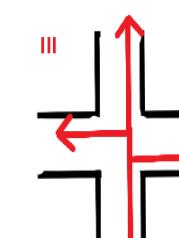
NO TIMER EXPIRED CONTINUE NORMAL FLOW

CURRENT STATE:
TIMER_E: 40 sec
TIMER_W: 10 sec
TIMER_N: 120 sec
TIMER_S: 0 sec



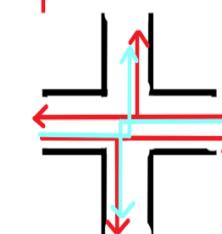
S TIMER EXPIRED SO WE MOVED
TO CORRESPONDING PHASE: PHAZE III
END GREEN
LIGHT PHAZE

CURRENT STATE:
TIMER_E: 0 sec
TIMER_W: 0 sec
TIMER_N: 20 sec
TIMER_S: 120 sec



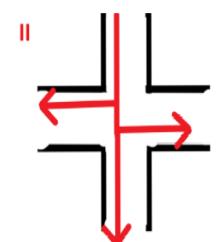
Conflict transițional

CURRENT STATE:
I
TIMER_E: 120 sec
TIMER_W: 120 sec
TIMER_N: 120 sec
TIMER_S: 120 sec



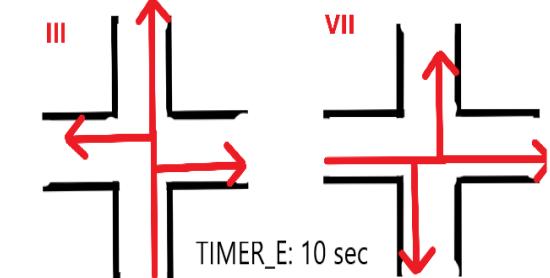
END GREEN
LIGHT PHAZE

CURRENT STATE:
II
TIMER_E: 120 sec
TIMER_W: 120 sec
TIMER_N: 10 sec
TIMER_S: 5 sec



NO TIMER EXPIRED CONTINUE NORMAL FLOW

POSSIBLE STATES



TIMER_E: 10 sec
TIMER_W: 0 sec
TIMER_N: 120 sec
TIMER_S: 0 sec

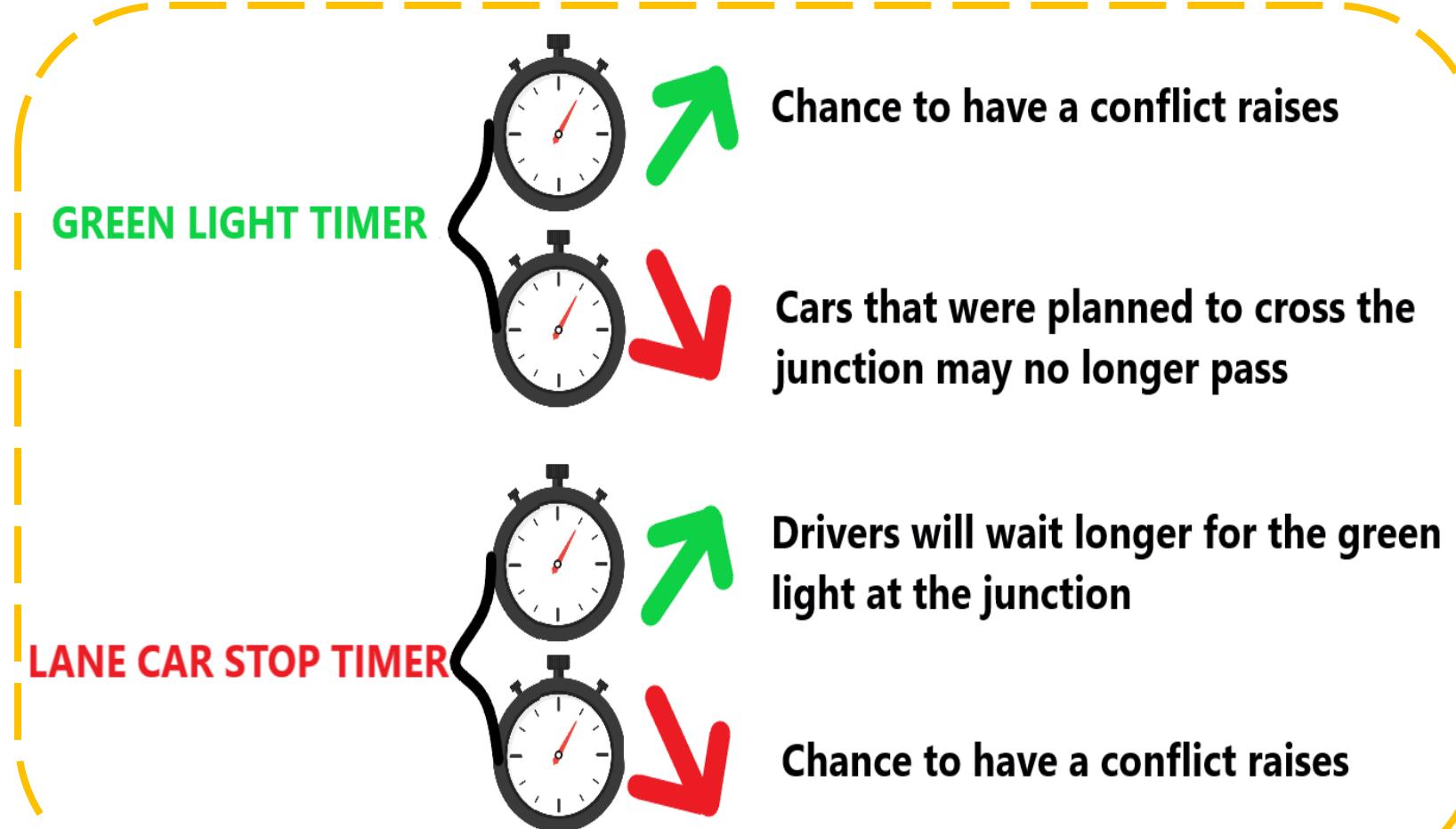
WE HAVE BOTH TIMER FOR E AND S EXPIRED
BUT NO CORRESPONDING PHAZE.

Influența traficului

- Adaptăm durata de roșu/verde pentru a evita stări de conflict
- Odată cu sosirea unui vehicul, cronometrul corespunzător este decrementat
- Impact diferit pentru fiecare client în parte
- Control asupra numărului de vehicule ce trec de intersecție



Impactul modificării cronometrelor



Control al traficului

X cars passed + **no conflict** → **LANE**



<X cars passed + **conflict** → **LANE**



X cars passed +
conflict

→ **GREEN LIGHT**



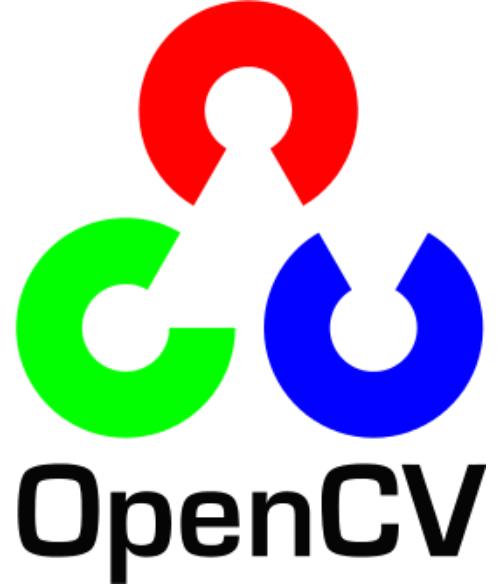
<X cars passed +
no conflict

→ **GREEN LIGHT**

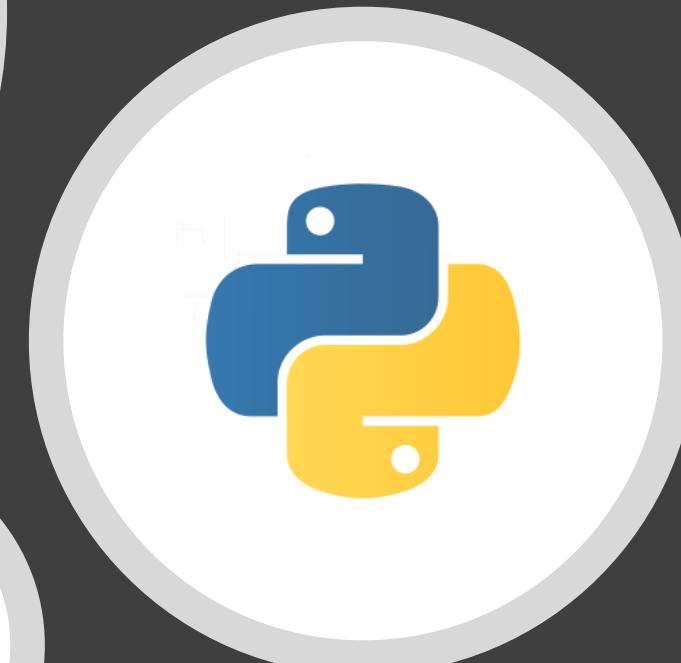


A vibrant, abstract graphic on the left side of the slide features a dense cluster of overlapping circles in various colors, including red, orange, yellow, green, blue, and purple. These circles vary in size and opacity, creating a dynamic, painterly effect that resembles a splash or a burst of energy. The background behind the circles is a lighter, more uniform color.

Detalii de implementare



Tehnologii folosite



The screenshot shows the Microsoft Visual Studio IDE interface. The top menu bar includes File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help, Search (Ctrl+Q), and TrafficManager. The solution explorer on the left lists nine projects under 'Solution Explorer (Ctrl+)': CarDetector, Header Files, Resource Files, Source Files, Common, data, db, utile, and GUIFW. The code editor window displays C++ code for 'proxy_config_data' and 'getData' functions. The status bar at the bottom indicates 'No issues found'.

```
struct JMSConfig
{
    std::map<LANE, std::string> laneToKeyword;
    uint8_t usingLeftLane;
    uint16_t maxWaitingTime;
    utile::IP_ADDRESS serverIp;
    utile::PORT serverPort;
    std::optional<LANE> missingLane = {};
};

struct proxy_config_data
{
    IP_ADDRESS ip = "";
    PORT port = 0;
    GeoCoordinate<DecimalCoordinate> boundSW{0., 0.};
    GeoCoordinate<DecimalCoordinate> boundNE{0., 0.};
    std::string dbServer = "";
    std::string dbUsername = "";
    std::string dbPassword = "";

    std::string toString() const
    {
        std::stringstream ss;
        ss << "ip= " << ip << " port= " << port << " "
           " boundSW: " << boundSW.toString() << " "
           " boundNE: " << boundNE.toString();
        return ss.str();
    }
};

template<typename T>
bool getData(const nlohmann::json& json, const std::string key)
{
    auto val = json.find(key);
    if (val == json.end())
    {
        std::cerr << "\\" + key + "\\ key missing";
        return false;
    }

    try
    {
        T value = val.get<T>();
        return true;
    }
    catch (const std::exception& e)
    {
        std::cerr << "Error parsing " + key + ": " << e.what();
        return false;
    }
}
```

Structură a proiectului

- **Librări statice:**

- Common
- Car Detector
- IPC
- GUI

- **Executabile:**

- Proxy
- Junction Main Server
- Object Detection Server
- Vehicle Tracker
- Traffic Observer
- Object Detection Server
- Testing

Videoclip(TO DO)

Concluzii și direcții viitoare

- **Dificultăți**

- determinarea formulei de actualizare a duratei cronometrelor
- colectarea de date + simularea traficului

- **Defecte ale sistemului**

- potențialul unui atac cibernetic la nivel de intersecție
- performanța afectată în condiții meteo defavorabile

- **Îmbunătățiri de viitor**

- adăugarea unui mecanism de “blacklisting”
- menținerea evidenței mașinilor la nivel geografic
- îmbunătățirea modelului antrenat