



Politechnika Wrocławska
Wydział Informatyki i Telekomunikacji

TRAFFIC ANALYZER

Aplikacja do analizy i wymiarowania ruchu telekomunikacyjnego

Przedmiot: Inżynieria Ruchu Teleinformatycznego

Autor: [Żukowski Illia, Zablotskyi Dmytro]
Nr albumu: [281302]

Wrocław, Styczeń 2026

Spis treści

1 Wstęp i cel projektu	2
1.1 Tło problemu	2
1.2 Cel aplikacji	2
2 Podstawy teoretyczne inżynierii ruchu	2
2.1 Definicja natężenia ruchu (Erlang)	2
2.2 TCBH vs FDMH - Istota problemu	3
2.2.1 FDMH (Fixed Daily Measurement Hour)	3
2.2.2 TCBH (Time Consistent Busy Hour)	3
2.3 Analiza Wiarygodności (Przedział Ufności)	3
3 Architektura systemu i technologie	4
3.1 Wykorzystane biblioteki	4
3.2 Struktura klas	4
4 Szczegółowy opis algorytmów	5
4.1 Wczytywanie i normalizacja danych	5
4.2 Algorytm Sliding Window (Wyznaczanie TCBH)	5
4.3 Algorytm Selekcji Czasowej (Time Gating)	5
5 Implementacja interfejsu (GUI)	6
5.1 Wizualizacja Danych	6
6 Instrukcja obsługi	6
6.1 Analiza danych rzeczywistych (Tryb Inżynierski)	6
6.2 Tryb Symulacji (Tryb Edukacyjny)	7
7 Wnioski	7

1 Wstęp i cel projektu

1.1 Tło problemu

Współczesne sieci telekomunikacyjne (GSM/LTE/5G oraz sieci VoIP) charakteryzują się dużą zmiennością obciążenia w czasie. Kluczowym wyzwaniem dla operatora jest takie zwymiarowanie zasobów sieciowych (liczba kanałów rozmownych, przepustowość łączy styków operatorskich), aby zapewnić wymaganą jakość usług (QoS - Quality of Service) przy jednoczesnej optymalizacji kosztów (CAPEX/OPEX).

Projektowanie sieci na podstawie średniej dobowej prowadzi do rażących błędów, ponieważ ruch w godzinach nocnych drastycznie zaniża średnią. Z kolei projektowanie na podstawie absolutnego maksimum (piku) z całego roku jest ekonomicznie nieuzasadnione.

1.2 Cel aplikacji

Celem niniejszego projektu było stworzenie oprogramowania narzędziowego **Traffic Analyzer**, które automatyzuje proces wyznaczania **Godziny Największego Ruchu (GNR)** zgodnie ze standardami ITU-T (International Telecommunication Union). Aplikacja ma za zadanie:

- Agregować dane z wielu dni pomiarowych (pliki CSV).
- Wyznaczać wskaźniki inżynierskie: TCBH, ADPH oraz FDMH.
- Szacować wiarygodność pomiaru poprzez analizę statystyczną (przedziały ufności).
- Wizualizować profil ruchu w celu szybkiej oceny anomalii.

2 Podstawy teoretyczne inżynierii ruchu

Analiza ruchu telekomunikacyjnego opiera się na statystyce matematycznej oraz teorii teletraffic. W programie zaimplementowano metodyki zgodne z rekomendacjami ITU-T (seria E.490 - E.500).

2.1 Definicja natężenia ruchu (Erlang)

Podstawową jednostką miary w systemie jest Erlang (Erl). Jeden Erlang oznacza ciągłe zajęcie jednego kanału rozmownego przez godzinę.

$$A = \frac{c \cdot h}{T} \quad (1)$$

Gdzie:

- A – natężenie ruchu [Erl].

- c – liczba wywołań w czasie obserwacji.
- h – średni czas trwania połączenia (Hold Time).
- T – czas obserwacji (w programie $T = 60$ minut).

2.2 TCBH vs FDMH - Istota problemu

Kluczowym zagadnieniem w wymiarowaniu sieci jest wybór odpowiedniej definicji godziny szczytu. Program rozróżnia dwa podejścia:

2.2.1 FDMH (Fixed Daily Measurement Hour)

To metoda "zegarowa", analizująca ruch w sztywnych ramach (np. 10:00-11:00). Jest prosta obliczeniowo, ale obarczona błędem systematycznym – jeśli rzeczywisty szczyt występuje o 10:30, to zostanie on "rozmyty" pomiędzy dwie godziny zegarowe. Prowadzi to do niedoszacowania potrzebnych zasobów.

2.2.2 TCBH (Time Consistent Busy Hour)

Metoda zaimplementowana w programie jako referencyjna. Polega na znalezieniu takich samych 60 minut dla wszystkich dni pomiarowych, dla których suma ruchu jest największa.

Algorytm Sliding Window: Program nie analizuje tylko pełnych godzin, ale przesuwa 60-minutowe okno z krokiem 1-minutowym po uśrednionym profilu.

$$TCBH_{val} = \max_{t \in \langle 0, 1380 \rangle} \left(\frac{1}{N} \sum_{d=1}^N A_d(t, t + 60) \right) \quad (2)$$

Gdzie N to liczba dni, a $A_d(t, t + 60)$ to natężenie ruchu w dniu d w oknie zaczynającym się w minucie t .

2.3 Analiza Wiarygodności (Przedział Ufności)

Samo wyznaczenie średniej wartości TCBH jest niewystarczające, gdyż ruch telekomunikacyjny jest procesem stochastycznym. Aby ocenić ryzyko, program wyznacza przedział ufności.

Ze względu na ograniczoną liczebność próby ($N \approx 30$ dni) oraz nieznane odchylenie standardowe populacji, zastosowano **rozkład t-Studenta**.

Wzór na granice przedziału ufności dla poziomu istotności $\alpha = 0.05$ (95%):

$$\mu \in \left(\bar{x} - t_{\alpha/2, n-1} \cdot \frac{S}{\sqrt{n}}; \bar{x} + t_{\alpha/2, n-1} \cdot \frac{S}{\sqrt{n}} \right) \quad (3)$$

Gdzie:

- \bar{x} – estymator średniej (wyliczone TCBH).
- S – odchylenie standardowe z próby w danej godzinie szczytu.
- n – liczba dni pomiarowych.
- $t_{\alpha/2,n-1}$ – kwantyl rozkładu t-Studenta.

Interpretacja inżynierska:

- **Wąski przedział:** Ruch jest stabilny i powtarzalny (charakterystyczne dla firm/biur).
- **Szeroki przedział:** Ruch jest chaotyczny. Należy zaprojektować sieć z dużym zaspasem (over-provisioning), aby uniknąć blokad.

3 Architektura systemu i technologie

Aplikacja została zrealizowana w paradygmacie programowania obiektowego (OOP) w języku **Python 3.10+**.

3.1 Wykorzystane biblioteki

CustomTkinter Nowoczesna biblioteka GUI. Zapewnia obsługę trybu ciemnego (Dark Mode) oraz skalowanie DPI na systemach macOS/Windows.

Pandas Biblioteka do manipulacji danymi. Używana do wczytywania plików CSV, normalizacji separatorów i operacji na macierzach.

NumPy Wykorzystywana do szybkich obliczeń wektorowych (np. splot `convolve`).

Matplotlib Silnik renderujący wykresy osadzony w oknie aplikacji.

SciPy Moduł wykorzystany do obliczania kwantyli rozkładu t-Studenta.

3.2 Struktura klas

Program podzielono na dwie główne klasy (wzorzec MVC):

1. **Klasa TrafficEngine (Model):** Odpowiada za logikę biznesową. Przechowuje surowe dane w liście ramek danych i wykonuje obliczenia statystyczne. Nie posiada zależności od GUI.
2. **Klasa TrafficApp (View/Controller):** Inicjalizuje okno, obsługuje zdarzenia użytkownika i rysuje wykresy.

4 Szczegółowy opis algorytmów

4.1 Wczytywanie i normalizacja danych

Aplikacja musi obsługiwać pliki pochodzące z różnych systemów bilingowych. Zaimplementowano mechanizm automatycznej detekcji separatora (średnik vs przecinek) oraz konwersji liczb zmiennoprzecinkowych (kropka vs przecinek).

```
1 # Przykład obsługi separatorów i konwersji typów
2 df = pd.read_csv(full_path, sep=';')
3 if 'ruch_erl' not in df.columns:
4     df = pd.read_csv(full_path, sep=',') # Fallback
5
6 # Konwersja formatu europejskiego (przecinek) na US (kropka)
7 df['ruch_erl'] = pd.to_numeric(
8     df['ruch_erl'].astype(str).str.replace(',', '.'),
9     errors='coerce'
10 ).fillna(0)
```

Listing 1: Fragment metody wczytującej dane

4.2 Algorytm Sliding Window (Wyznaczanie TCBH)

Zamiast iteracji pętlą, użyto operacji splotu (konwolucji), co jest rozwiązaniem optymalnym obliczeniowo.

```
1 # 1. Obliczenie średniego profilu ze wszystkich dni
2 sredni_profil = macierz.mean(axis=1)
3
4 # 2. Zastosowanie okna przesuwnego (splot dyskretny)
5 okno = 60 # 60 minut
6 srednia_ruchoma = np.convolve(sredni_profil, np.ones(okno)/okno, mode='valid')
7
8 # 3. Znalezienie indeksu maksymalnego
9 self.tcbh_start_index = idx_start + np.argmax(srednia_ruchoma)
```

Listing 2: Obliczanie TCBH metodą splotu

4.3 Algorytm Selekcji Czasowej (Time Gating)

Jedną z kluczowych funkcjonalności aplikacji jest możliwość zawężania zakresu analizy (np. od 8:00 do 20:00).

Uzasadnienie biznesowe: W sieciach występują dwa rodzaje ruchu: H2H (ludzki) i M2M (maszynowy, np. nocne backupy). Automatyczne procesy nocne mogą generować sztuczne "piki". Zwymiarowanie łącza głosowego pod nocny backup danych jest błędem inżynierskim.

Implementacja: Zrealizowano to poprzez "wycinanie" (slicing) macierzy danych przed przystąpieniem do szukania maksimum.

```

1 # Konwersja godzin (int) na indeksy minutowe (0-1440)
2 idx_start = max(0, int(start_h * 60))
3 idx_end = min(1440, int(end_h * 60))
4
5 # Ekstrakcja wycinka profilu
6 profil_analizowany = sredni_profil[idx_start:idx_end]
7
8 # Dalsza analiza odbywa się TYLKO na wyciętym fragmencie
9 local_max_idx = np.argmax(srednia_ruchoma)

```

Listing 3: Logika zawężania okna analizy

5 Implementacja interfejsu (GUI)

Interfejs podzielono na trzy sekcje funkcjonalne: panel sterowania (import i parametry), dashboard wyników (karty) oraz obszar wykresu.

5.1 Wizualizacja Danych

Wykres wykorzystuje bibliotekę Matplotlib w stylu "Flat Design", dopasowanym do ciemnego motywu aplikacji.

- **Linie szare (alpha=0.15):** Rzeczywiste przebiegi z poszczególnych dni. Ich niska przezroczystość tworzy "mapę ciepła", pokazującą gęstość próbek.
- **Linia cyjanowa:** Średni profil miesięczny (model).
- **Obszar czerwony:** Wizualizacja wyznaczonego okna TCBH (60 minut).
- **Pionowe linie żółte:** Granice analizy (filtry czasowe).

6 Instrukcja obsługi

6.1 Analiza danych rzeczywistych (Tryb Inżynierski)

1. Uruchom program `TrafficAnalyzer.app`.
2. W sekcji "Parametry Analizy" wybierz zakres godzin (np. OD: 8, DO: 16), aby wyeliminować anomalie nocne.
3. Kliknij przycisk "**Wgraj Folder z Pomiarami**".
4. Wskaż katalog zawierający pliki CSV (jeden plik na jeden dzień).
5. Odczytaj wynik TCBH z panelu Dashboard.
6. Sprawdź kolor karty "Przedział Ufności" (Zielony = wynik pewny, Czerwony = duża zmienność).

6.2 Tryb Symulacji (Tryb Edukacyjny)

Kliknięcie przycisku "**Symulacja (Auto)**" uruchamia generator Monte Carlo. Program tworzy 31 wirtualnych dni pomiarowych z wykorzystaniem rozkładu wielomianowego i szumu Gaussa. Pozwala to przetestować działanie algorytmów bez posiadania zewnętrznych danych.

7 Wnioski

Zrealizowany projekt spełnia wszystkie założenia specyfikacji. Aplikacja pozwala na szybką analizę dużych zbiorów danych bez konieczności manualnej obsługi arkuszy kalkulacyjnych.

Zastosowanie "okna przesuwnego" pozwala na znacznie dokładniejsze wyznaczenie szczytu niż prosta metoda zegarowa (FDMH), co może przynieść operatorowi wymierne oszczędności przy zakupie pasma. Dodatkowo, wdrożenie modułu analizy statystycznej wprowadza kluczowy w inżynierii element oceny ryzyka.

Aplikacja jest gotowa do wdrożenia w środowisku produkcyjnym jako narzędzie wspomagające pracę działu planowania sieci.

Literatura

- [1] ITU-T Recommendation E.492, *Traffic reference period*, International Telecommunication Union.
- [2] Iversen V.B., *Teletraffic Engineering and Network Planning*, Technical University of Denmark, 2010.
- [3] Dokumentacja biblioteki Pandas, <https://pandas.pydata.org/>
- [4] Virtanen P. et al., *SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python*, Nature Methods, 2020.