VŠB TECHNICKÁ UNIVERZITA OSTRAVA
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Provozní charakteristiky volání v korporátním prostředí

Traffic Characteristics in Corporate Environment

Adam Truchlý

Bakalářská práce

Vedoucí práce: prof. Ing. Miroslav Vozňák, Ph.D.

Ostrava, 2021

Abstrakt

Cieľom práce je analýza dát z pobočkovej telefónnej ústredne VŠB-TUO za časové obdobie desiatich rokov. V prvej časti práce sú popísané teoretické poznatky o obsluhe a teória využitých funkcií. Praktická časť mojej práce sa zameriava na vytvorenie programu v programovacom jazyku Python, ktorý spracuváva dataset, umožní tým jeho detailnú analýzu a aplikáciu modelu Erlang B a Engset. Práca obsahuje radu analýz a rôznych štatistík týkajúcich sa prevádzky pobočkovej ústredne.

Kľúčové slová

analýza dát; Python; CDR; Erlang B; Engset

Abstract

The aim of this thesis is to analyse data over the period 10 years from the VSB-TUO private branch exchange. The first section focuses on theoretic basics of queuing theory. The practical aspect of my work aims to create a program in the Python programming language. It processes a dataset, which enables a deeper analysis of the dataset with applying models Erlang B and Engset. The thesis includes a detailed analysis of the private branch exchange operation as well as other statistics concerning it.

Keywords

data analysis; Python; CDR; Erlang B; Engset

Poďakovanie

Rád by som sa poďakoval vedúcemu bakalárskej práce, prof. Ing. Miroslav Vozňák, Ph.D., za účinnú metodickú a odbornú pomoc, konzultácie a cenné rady pri spracovaní mojej bakalárskej práce.

Obsah

Zoznam použitých skratiek a symbolov

CDR	–	Call Detail Records
FCFS	–	First Come, First Served
FIFO	–	First In, First Out
LCFS	–	Last Come, First Served
LIFO	–	Last In, First Out
RSA	–	Rivest–Shamir–Adleman
VŠB-TUO	–	Vysoká škola baňská - Technická univerzita Ostrava

Kapitola 1

Predstavenie záverečnej práce

Témou tejto bakalárskej práce je zanalyzovať reálne dáta z pobočkovej telefónnej ústredne VŠB - TUO za obdobie desiatich rokov. Tieto dáta obsahujú údaje, ktoré by mohli viesť k úniku citlivých informácií a to konkrétne údaje ako telefonné čísla, či čas hovoru. Z tohto dôvodu bolo potrebné aplikovať anonymizáciu dát. Anonymizácia dát bola prevedená pomocou aplikácie, ktorá bola v rámci praktickej časti práce vytvorená. Samotná aplikácia taktiež generuje grafy, spracúva informácie z datasetu o prevádzke a vyhodnocuje ju.

Práca je štrukturovaná nasledovne, v nasledujúcej kapitole sa zoznámime s úvodom do teórie hromadnej obsluhy, kde sa popisuje základný matematický súhrn a predstavujú nástroje, ktoré sa v oblasti prevádzkového zaťaženia využívajú. V ďalšej kapitole je popísaný CDR záznam, kde sú zároveň detailne vysvetlené jeho časti. Potom sú popísané hašovacie funkcie, ktoré boli implementované v rámci práce z dôvodu anonymizácie.

Štvrtá kapitola sa zameriava na praktickú časť. Týka sa implementácie anonymizácie, prácu s programom a demonštruje konverziu dát, ktorá bola potrebná v rámci spracovania dát. Posledná kapitola zobrazuje a zhrnuje dosiahnuté výsledky, ktoré sa podarili docieliť pomocou navrhnutej aplikácie. Dosiahnuté výsledky umožňujú aplikáciu modelov Erlang B, Engset a následné porovnanie oboch modelov.

Kapitola 2

Úvod do teórie hromadnej obsluhy

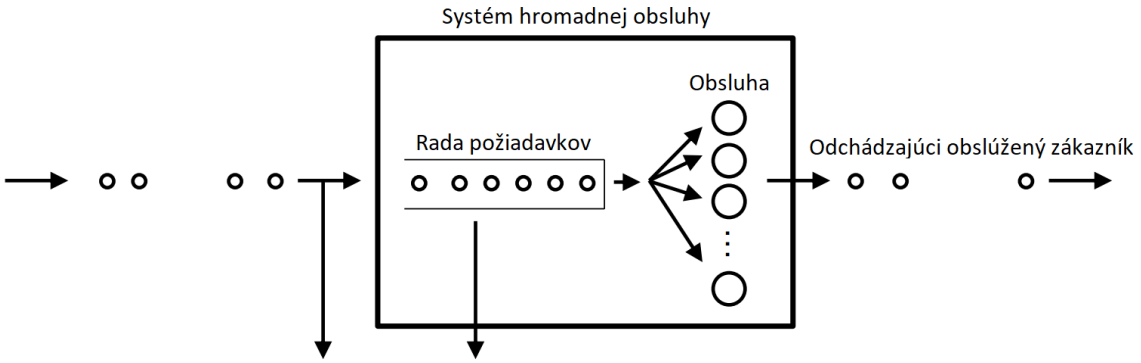
Začiatkom 20. storočia začal vzrastať počet používateľov telefónnych zariadení. Zvýšený počet používateľov telefónnych zariadení ústil k zvýšenému počtu uskutočnených hovorov čo sa premietlo v požiadavke na kapacitu telekomunikačnej siete. Vzhľadom na stavebné náklady na stavbu siete pri dlhých vzdialenostiach, zlý odhad očakávanej prevádzky a predimenzovanie kapacity by mohli viesť k ekonomickým problémom s návratnosťou investície. Z iného uhla pohľadu by poddimenzovanie kapacity mohlo viesť k strate zisku a nespokojnosti zákazníkov. Preto analýza závislostí telekomunikačnej prevádzky sa stal dôležitou časťou výskumu. S touto problematikou sa zaoberal dánsky matematik Erlang, ktorý využil riešenie pomocou teórie hromadnej obsluhy.

Mnoho inžinierov a matematikov sa snažilo na tento problém použiť metódy pravdepodobnosti, ktoré neboli dostatočne použiteľné na uvedenú problematiku. Teória hromadnej obsluhy sa veľmi rýchlo stala odvetvím aplikovanej matematiky a mnoho výsledkov je dodnes používaných v oblasti počítačov, teórie spoľahlivosti, telekomunikáciach či dopravnom inžinierstve.

Tento systém je úspešný na vyhľadávanie slabín v odvetví poskytovania služieb. Laicky pod teóriou hromadnej obsluhy môžeme rozumieť automobil čakajúci na svetelnej križovatke na prejazd, zákazníka v reštaurácii čakajúci na obsluhu alebo telefónne spojenie účastníka v ústredni a mnoho ďalších [1] [2].

2.1 Systém hromadnej obsluhy

Obrázok 2.1 zobrazuje systém hromadnej obsluhy. Proces pozostáva z príchodu zákazníka, čakania v rade, obslúžení a odchodu. Môže ale dôjsť aj k prípadu, kedy zákazník z rady vystúpi predčasne pred obslúžením, prípadne ešte pred vstupom do rady.



Obr. 2.1: Príklad procesu hromadnej obsluhy [3]

2.2 Primárne parametre

V množstve prípadov sa dá charakterizovať oblužný systém vo všeobecnosti šiestimi krokmi: proces príchodu zákazníka, vzor obsluhy, čakanie v rade, kapacita systému, počet obslužných kanálov a fázu obsluhy [3] [4] [5].

- **Proces príchodu zákazníka:**

Za normálnych okolností je proces príchodu čiste náhodný – stochastický a nezávislý na čase. Pre optimalizáciu procesov je potrebné vypozorovať nasledovné: časy príchodu zákazníkov, potom možnosť, či zákazníci môžu doraziť súčasne a ďalej faktor netrpezlivosti zákazníka vysvetlený nasledovne.

Zákazník sa môže rozhodnúť nevstupovať do systému v prípade, ak je rada príliš dlhá. Naopak môže počkať bez ohľadu na to ako dlho bude v rade čakať. Ak už v rade čaká, môže nastať prípad, že stratí trpezlivosť a rozhodne sa odísť. Ak existujú dva alebo viac paralelných čakacích radov, zákazník môže zmeniť svoj rad v ktorom stojí.

- **Vzor obsluhy:**

Vzor obsluhy popisuje službu ako jednoduchú alebo hromadnú. Pri jednoduchej obsluhe jeden systém obsluhuje jedného zákazníka. Hromadná obsluha ale môže obsluhovať viacerých zákazníkov v jeden okamih a tým sa stáva efektívnejšia. Napríklad návštevníci prehliadky so sprievodcom.

- **Čakanie v rade:**

Existuje viacero možností čakania v rade. Ak systém nemá predurčenú možnosť čakania, tak sa typicky jedná o FCFS (first come first serve) – prvý prichádzajúci je zároveň prvý obslúžený. Iný rad LCFS (last come, first serve) - prichádzajúca posledná požiadavka je prvá obslúžená. Tento systém

je využívaný na dosiahnutie bližšej položky v inventárnom systéme. Ďalej náhodný výber vyberá zákazníkov náhodne a nezávisle od času príchodu. V prioritnej schéme výberu sa upredňostňuje zákazník s vyššou prioritou voči zákazníkovi s nižšou. V prípade ak už je zákazník s nižšou prioritou obsluhovaný, musí zákazník s vyššou prioritou počkať na obslúženie.

- **Kapacita systému:**

Kapacita systému predstavuje koľko zákazníkov môže byť obsluhovaných v systéme. Túto situáciu predstavuje preplnenené metro. Cestujúci musí počkať na fyzické uvoľnenie miesta prípadne na ďalšiu súpravu metra.

- **Počet obslužných kanálov:**

Kompromis medzi počtom obslužných kanálov a oneskorení zákazníka popisuje číslo servisných kanálov. Pridaním obsluhy vzniknú náklady ale zároveň sa podstatne zníži čakanie zákazníka. Cieľom je nájsť optimálny stav medzi kvalitou poskytovaných služieb a nákladmi na dosiahnutie kvality služby.

- **Fázy obsluhy:**

Systém obsluhy môže mať jednu fázu alebo niekoľko fáz. Príkladom pre viac fázovú obsluhu môže byť vyšetrenie u všeobecného lekára, ktorý nás pošle na vyšetrenie ucha, krvné testy, vyšetrenie oka.

2.3 Merateľné parametre

Rôzne typy systémov čakania v rade sú matematicky analyzované za účelom stanovenia výkonu hodnôt z popisu systému. Pretože model radenia do fronty predstavuje dynamický systém, hodnoty meracích ukazovateľov sa menia v čase. Výkonostné merateľné parametre potom podľa [4] [6] sú:

- **Pravdepodobnosť počtu úloh v systéme π_k:**

Správanie systému čakania v rade je často možné opísať pomocou vektoru pravdepodobnosti počtu úloh v systéme π_k. Priemerná hodnota väčšiny ďalších výkonnostných ukazovateľov je možná odvodiť z π_k:

$$\pi_k = P[k \text{ úloh } v \text{ systéme}].$$

- **Zaťaženie systému A:**

Ak systém čakania vo fronte pozostáva z jedného radu, potom zaťaženie systému A je zlomok času kedy je rad obsadený. V prípade, že nie je obmedzený počet úloh v systéme jednoduchých front je potom zaťaženie fronty dané vzťahom :

$$A = \frac{priemerný\ čas\ strávený\ obsluhou}{priemerný\ čas\ príchodu} = \frac{miera\ príchodu}{miera\ služby} = \frac{\lambda}{\mu}. \tag{2.1}$$

Využitie systému hromadnej obsluhy s viacerými radmi sa rovná priemernej hodnote aktívnych front. Keďže $c\mu$ je celková miera služieb, potom máme :

$$A = \frac{\lambda}{c\mu}, \tag{2.2}$$

a A môže byť použité na formulovanie podmienky pre stacionárne správanie vyššie spomenuté. Podmienkou stability je :

$$A < 1; \tag{2.3}$$

kde v priemere musí byť počet požiadaviek, ktoré prídu za jednotku času menší ako počet požiadaviek, ktoré je možné spracovať.

- **Intenzita požiadaviek λ:**

Intenzita je v elementárnom systéme front definovaná ako priemerný počet požiadaviek, ktorých spracovanie je dokončené v jednej časovej jednotke, tzv. rýchlosť odchodu. Pretože rýchlosť odchodu je rovnaká miere príchodu λ pre systém obsluhy v štatistickej rovnováhe potom je intenzita daná vzťahom:

$$\lambda = m \cdot A \cdot \mu, \tag{2.4}$$

kde m reprezentuje počet agentov, A zaťaženie systému a μ mieru služby.

- **Doba odozvy T:**

Doba odozvy, známa tiež ako čas pobytu je celkový čas, ktorý úloha strávi v systéme radenia do fronty.

- **Čakacia doba W:**

Čakacia doba je čas, ktorý úloha strávi v rade čakaním na obsluhu. Preto máme:

$$\textit{Doba odozvy} = \textit{čakacia doba} + \textit{doba služby}.$$

Pretože W a T sú zvyčajne náhodné čísla, počíta sa ich priemer. Potom:

$$\bar{T} = \bar{W} + \frac{1}{\mu}. \tag{2.5}$$

- **Dĺžka fronty Q:**

Dĺžka fronty Q značí počet požiadaviek čakajúcich na obsluhu.

- **Počet požiadavok v systéme K:**

Počet požiadavok v systéme front je označovaný ako K. Potom:

$$\bar{K} = \sum_{k=1}^{\infty} k \cdot \pi_k. \tag{2.6}$$

Priemerný počet požiadavok v systéme front $\bar{K}$ a priemerná dĺžka fronty $\bar{Q}$ môže byť vypočítaná pomocou Littleho teorému:

$$\bar{K} = \lambda + \bar{T} \tag{2.7}$$

$$a\ \bar{Q} = \lambda + \bar{W} \tag{2.8}$$

2.4 Littleho vzťah

Littleov vzťah nám stanovuje veľmi dôležitý pomer využívaný v systéme hromadnej obsluhy. Poskytuje vzťah medzi troma hodnotami, ktorý hovorí o priemernom počte požiadavkov v systéme. Očakávaný priemerný počet požiadavkov v systéme L sa rovná súčinu priemernej hodnoty príchodu požiadavkov vstupujúcich do systému λ a priemerne očakávané množstvo času, ktorý požiadavka v systéme strávi W [4]. Potom:

$$L = \lambda \cdot W \tag{2.9}$$

Uvažujeme tak o systéme, v ktorom sú požiadavky obsluhované podľa poradia príchodu za predpokladu, že počet požiadavok nerastie do nekonečna. Teda predpokladáme, že je dostatočná kapacita systému pre spracovanie požiadaviek.

Príklad. Máme systém dátovej komunikácie s troma prenosovými linkami. Pakety dorazia na tri rôzne uzly s rýchlosťou príchodu $\lambda_1 = 200$ paketov/s, $\lambda_2 = 300$ paketov/s a $\lambda_3 = 10$ paketov/s. Predpoklad je, že v tomto systéme prechádza v priemere 50 000 paketov podobnej veľkosti [7].

Potom použitím Littleho vzťahu je priemerné oneskorenie na paket v systéme:

$$W = \frac{L}{\lambda_1 + \lambda_2 + \lambda_3} = \frac{50000}{200 + 300 + 10} = 98,4s. \tag{2.10}$$

2.5 Markovovské modely systémov hromadnej obsluhy

Markovove procesy reprezentujú silné a efektívne prostriedky pre analýzu a popis systémových vlastností. Procesy zároveň zahrňujú neurčitosť a náhodné javy. Všeobecný príklad Markovovho

procesu môže byť model hier, kde sa hádže kockou, program v ktorom sa generuje náhodné číslo alebo správanie ľudí v supermarkete.

Pre tieto procesy platí, že z akéhokoľvek stavu je možné prechádzať s určitou pravdepodobnosťou do jedného alebo viacerých stavov. Stav, do ktorého sa prechádza, môže byť podobný stavu predchádzajúcemu avšak za podmienky ktorá stanovuje, že súčet pravdepodobností prechodu z každého stavu je rovný jednej. Ďalšia vlastnosť, ktorá musí byť splnená je tzv. Markovova vlastnosť. Táto vlastnosť požaduje, aby rozhodnutia, ktoré sú učinené v každom stave boli založené práve na tom, o aký stav sa jedná. Teda nie je povolené aby rozhodnutie záviselo na predchádzajúcich udalostiach daného stavu. Zvláštnym typom Markovovského procesu je proces vzniku a zániku *birth and dead process*, v ktorom sa stav udáva v počte členov populácie a mení sa v čase [8] [9].

2.6 Poissonovo rozdelenie

Poissonovo rozdelenie je využívané v systémoch hromadnej obsluhy pre opis vstupného toku. Patrí medzi diskrétne rozdelenia a definuje sa funkciou, ktorá určuje pravdepodobnosť, s akou náhodná premenná X nadobúda hodnoty $k = 1, 2, ...$ s pravdepodobnosťou:

$$p_k = P[X = k] = \frac{\lambda^k}{k!} \cdot e^{-\lambda} \tag{2.11}$$

Stredná hodnota

$$E(X) = \sum_{k=0}^{\infty} k \cdot \frac{\lambda^k}{k!} \cdot e^{-\lambda} = \lambda \tag{2.12}$$

Rozptyl

$$D(X) = E(X^2) - [E(X)]^2 = \lambda^2 + \lambda - \lambda^2 = \lambda \tag{2.13}$$

Parameter Poissonovho rozdelenia λ je strednou hodnotou aj rozptylom náhodnej premennej s Poissonovým rozdelením. Náhodná premenná má využitie v mnohých aplikáciach a to napríklad pri aproximácii binomickej náhodnej premennej, za predpokladu, že má veľký parameter n (počet pokusov) a malý parameter p (úspešná pravdepodobnosť pokusu) [10].

2.7 Exponenciálne rozdelenie

Náhodná veličina X má pri exponenciálnom rozdelení pravdepodobnostné rozdelenie spojitého typu. Popisuje ju spojitá funkcia pravdepodobnosti f(x):

$$f(x) = \begin{cases} \lambda \cdot e^{-\lambda} ..... \; pre \; x > 0 \\ 0 ..... \; pre \; x \leq 0 \end{cases} . \tag{2.14}$$

Stredná hodnota

$$E(X) = \lambda \int_0^\infty x \cdot e^{-\lambda \cdot x}\, dx = \frac{1}{\lambda}. \tag{2.15}$$

Rozptyl

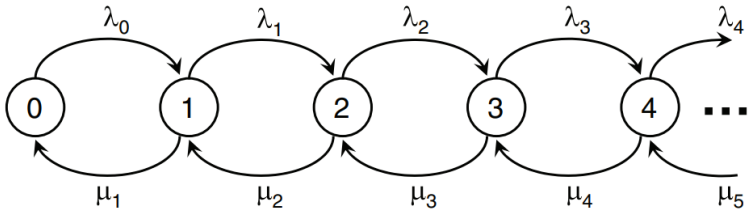
$$D(X) = \lambda \int_0^\infty (x - \frac{1}{\lambda})^2 \cdot e^{-\lambda \cdot x}\, dx = \frac{1}{\lambda^2}. \tag{2.16}$$

To znamená, že stredná hodnota je obrátenou hodnotou parametra 1. Rozptyl je druhou mocninou strednej hodnoty.

Exponenciálne a Poissonovo rozdelenie opisuje rovnakú podstatu z dvoch hľadísk. Zatiaľ čo v Poissonovom rozdelení sa preukazuje pravdepodobnosť výskytu určitého počtu javov za jednotku času, tak v exponenciálnom rozdelení sa udáva pravdepodobnosť dĺžky intervalu medzi dvoma javmi, ktoré nastali [10].

2.8 Proces vzniku a zániku v systéme hromadnej obsluhy

Proces vzniku a zániku pozostáva zo sady stavov (0, 1, 2, ...), ktoré sa označujú ako populácia systému. K prechodom medzi týmito stavmi dochádza, keď aktuálny stav sa zmení smerom hore alebo dolu. Teda keď je systém v stave $n \geq 0$, čas do ďalšieho príchodu (vzniku) je exponenciálna náhodná premenná s rýchlosťou λ_n. V čase odchodu sa systém presúva zo stavu n do stavu $n + 1$. Keď sa systém nachádza v stave $n \geq 1$ potom čas do ďalšieho odchodu (zániku) je exponenciálna náhodná premenná s rýchlosťou μ_n. Pri zániku sa systém dostáva zo stavu n do stavu $n - 1$.



Obr. 2.2: Prechodový diagram pre proces vzniku a zániku [3]

2.9 Kendallove notácie

Ako skratka pre popis procesov čakania v rade sa vyvinula notácia, ktorú predstavil D. G. Kendall v roku 1953. Proces radenia do fronty je úsporne popísaný radou symbolov a lomítok $A/B/X/Y/Z$, kde:

A - reprezentuje časové rozloženie prichádzajúcich požiadaviek do systémov hromadnej obsluhy.

B - predstavuje pravdepodobnostný typ rozdelenia popisujúci dobu trvania obsluhy.

X - číslo udávajúce paralelný počet kanálov obsluhy.
Y - je číslo predstavujúce kapacitu systému hromadnej obsluhy, ak nie je kapacita obmedzená, využíva sa symbol ∞ .
Z - popisuje disciplínu obsluhy tzv. poradie obsluhy (FIFO,LIFO).

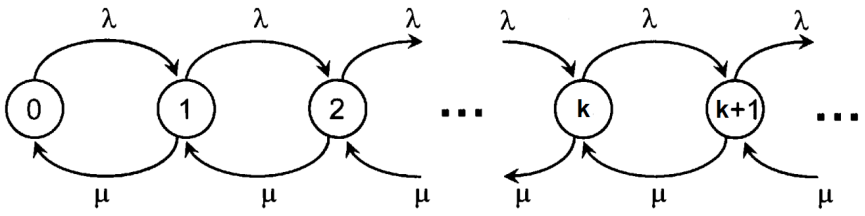
2.10 Markovovské modely

Markovovské modely majú značenie $M/M/X/Y$. Sú založené na nasledujúcich princípoch [11] :

- počet požiadaviek je väčší ako množstvo obslužných liniek,
- požiadavky sú generované náhodne a nezávisle na sebe,
- priemerný počet požiadaviek za jednotku času je konštantný pri všetkých zdrojoch,
- doba vybavenie požiadavky je náhodná premenná s exponenciálnym rozdelením,
- proces čakania je založený na algoritme FIFO.

2.11 Systém M/M/1/∞

Tento systém predpokladá jednu obslužnú linku s dobou obsluhy, ktorá sa riadi exponenciálnym rozdelením a taktiež predpokladá príchod požiadavkov do systému tiež pod exponenciálnym rozdelením. Je zložený z jedného serveru pripojeného k neobmedzenému počtu FIFO front. Ide potom o exponenciálne rozdelenie medzi príchodmi požiadavok s exponenciálnou intenzitou príchodu žiadostí λ. Doba obsluhy je exponenciálna s priemernou intenzitou obsluhy μ. Systém je stabilný za podmienky $\lambda < \mu$ [12].



Obr. 2.3: Grafické znázornenie reťazca M/M/1/∞

Z Obr. 3 možno vidieť, že $\lambda_k = \lambda$ pre $k \geq 0$ a $\mu_k = \mu$ pre $k \geq 1$. Potom pravdepodobnosť, že v systéme je k zákazníkov (obsluhovaných a zároveň v čakacej rade) je daná:

$$P_k = (1 - A) \cdot A^k \tag{2.17}$$

kde $A = \lambda/\mu$.

Náhodná premenná N je reprezentujúca počet požiadaviek v systéme potom platí:

$$N = \sum_{k=0}^{\infty} k \cdot p_k = \frac{A}{1-A}. \tag{2.18}$$

Priemerný čas, ktorý požiadavka strávi v systéme:

$$T = \frac{N}{\lambda} = (\frac{A}{1-A}) \cdot (\frac{1}{\lambda}) = \frac{\frac{1}{\mu}}{1-A}, \tag{2.19}$$

kde $\frac{1}{A}$ je obslužný čas pre jednu požiadavku.

Uvedený model slúži iba ako teoretický, keďže v skutočnosti nie je možné realizovať nekonečne dlhý rad. V praxi je ale možné sa mu čiastočne priblížiť s veľmi dlhým čakacím radom.

2.12 Systém M/M/∞/∞

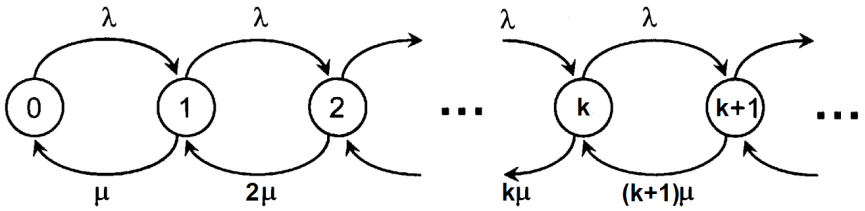
Systém je zložený z nekonečného počtu serverov a nekonečného radu pre ktorý platí:

$$\lambda_k = \lambda \; pre \; k \geq 0, \tag{2.20}$$

$$\mu_k = k \cdot \mu \; pre \; k \geq 1. \tag{2.21}$$

Potom je možné odvodiť rovnicu:

$$p_k = p_0 \cdot \prod_{i=0}^{k-1} \frac{\lambda}{(i+1) \cdot \mu} = \frac{(\frac{\lambda}{\mu})^k}{k!} \cdot e^{-\frac{\lambda}{\mu}} \; pre \; k \geq 0. \tag{2.22}$$



Obr. 2.4: Grafické znázornenie reťazca M/M/∞/∞

Priemerný počet požiadaviek v systéme je:

$$N = \frac{\lambda}{\mu}. \tag{2.23}$$

a priemerný čas strávený v systéme je úmerný priemernému času obsluhy jednej požiadavky:

$$T = \frac{1}{\mu}. \tag{2.24}$$

2.13 Systém M/M/m/∞

Systém pracuje s m obslužnými servermi a čakací rad je nekonečný. Požiadavky do systému prichádzajú Poissonovskou distribúciou λ náhodne a sú opísané exponenciálnym rozdelením μ. Preto je systém obsluhovaný rýchlosťou λ (rýchlosť prechodu zo stavu k do stavu $k+1$ kde k je počet požiadaviek v systéme). Na druhej strane je systém vyprázdňovaný rýchlosťou $k\mu$(stav $k+1$ do stavu k). Popísané správanie je platné do stavu $k \leq m$, pre prípad viacerých požiadaviek k v systéme ako je celkový počet agentov m, môžu agenti spracovávať požiadavky iba s rýchlosťou $m\mu$ a každá prichádzajúca požiadavka musí čakať v čakacej rade na voľného agenta. Táto rada má neobmedzenú kapacitu [13].

Parameter P_Q je reprezentujúci pravdepodobnosť, že volajúci zákazník bude musieť čakať v rade na obsluhu:

$$P_Q = \frac{(m \cdot A)^m}{m!(1-A)} \cdot P_0. \tag{2.25}$$

kde A označuje zaťaženie systému (2.1), platí podmienka stability (2.3), m je početnosť agentov a P_0 reprezentuje pravdepodobnosť systému, že sa v ňom nenachádza žiadna požiadavka je:

$$P_0 = \left\{ \sum_{k=0}^{m-1} \frac{(m)^k}{k!} + \frac{(m \cdot A)^m}{m!} \cdot \left(\frac{1}{1-A} \right) \right\}^{-1} \tag{2.26}$$

Priemerný počet požiadaviek v systéme je N:

$$N = A \cdot m + \frac{A}{(1-A)} \cdot P_Q, \tag{2.27}$$

priemerný čas požiadavky strávený v systéme T:

$$T = \frac{1}{\mu} + \frac{P_Q}{m \cdot \mu - \lambda}. \tag{2.28}$$

Priemerný počet požiadaviek v čakacom rade je Q:

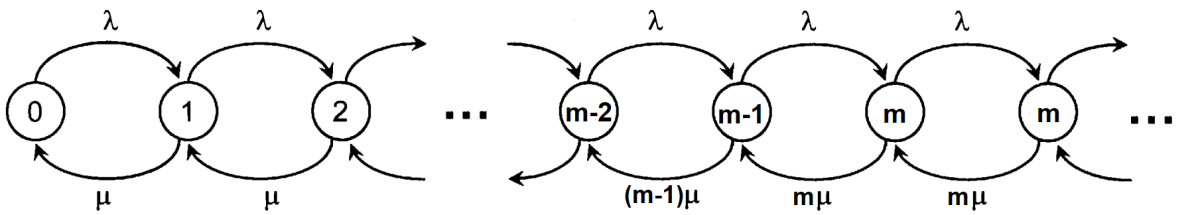
$$Q = \frac{A}{(1-A)} \cdot P_Q, \tag{2.29}$$

priemerný čas požiadavky strávený v čakacom rade W:

$$W = \frac{A}{1-\mu} \cdot \frac{P_Q}{\lambda}. \tag{2.30}$$

Z definície parametrov λ a μ máme:

$$A = \frac{\lambda}{m \cdot \mu}, \tag{2.31}$$



Obr. 2.5: Grafické znázornenie reťazca M/M/m/∞

2.14 Model Erlang - B

Erlang je bezrozmerná jednotka používaná v telefónii ako štatistická hodnota miery prevádzkového zaťaženia. Prevádzkové zaťaženie jedného Erlangu zodpovedá neustálemu obsadeniu jednej obsluhovej linky po dobu obsluhy.

Pre dimenzovanie telefónnych sietí bol počiatkom 20. storočia vyvinutý účinný nástroj v podobe Erlangových rovníc. Tieto rovnice sú aj v súčasnej dobe uznávaným a používaným nástrojom v oblasti prevádzkového zaťaženia. Erlang B rovnica dokáže určiť, s akou pravdepodobnosťou P_N budú všetky kanály obsadené, pokiaľ je známe zaťaženie A(2.1) a počet serverov N v spojovacom zväzku [14].

$$P_N = E_{1,N}(A) = \frac{\frac{A^N}{N!}}{\sum_{i=0}^{N} \cdot \frac{A^i}{i!}} \tag{2.32}$$

Rovnica musí splňovať podmienky:

- tok požiadavkov vzniká náhodne, s exponenciálnym rozložením odstupov medzi prichádzajúcimi susednými požiadavkami, čo znamená, že čím väčší odstup, tým menší je počet týchto prípadov.
- Doba obsluhy má podobné rozdelenie, to znamená že exponenciálne ubúda tých požiadavkov, ktoré majú dlhšiu dobu obsluhy.
- Tok požiadavkov je ustálený, teoreticky akoby vznikal z nekonečného počtu zdrojov požiadaviek.
- Existuje plná dostupnosť požiadavok na všetky obsluhové linky.

- Odmietnuté požiadavky sa nevracajú do vstupného toku, teda neexistujú opakované požiadavky. Tieto požiadavky obsadzujú systém na nulovú dobu.

- Nevznikajú súčasne dve požiadavky.

2.15 Engsetova rovnica

Engsetova rovnica zovšeobecňuje Erlangovu rovnicu tým, že definuje konečný počet volajúcich. Engsetova rovnica má tri vstupné parametre a to A - zaťaženie (2.1), c - počet serverov, N - počet zdrojov. Výstupom je pravdepodobnosť, že budú všetky linky obsadené P [15].

$$P = \frac{(\frac{N-1}{c}) \cdot (A)^c}{\sum_{i=0}^{c} (\frac{N-1}{i}) \cdot (A)^i} \tag{2.33}$$

Kapitola 3

Príprava dátovej sady a anonymizácia

Cieľom tejto kapitoly je čitateľa oboznámiť s funkciami ako je haš, podrobnosti záznamov o volaní či samotnú pobočkovú ústredňu z ktorej pochádzajú dáta, ktoré som spracovával. Podrobnejšie ozrejmujem hašovaciu funkciu, jej význam a druhy funkcií. V podkapitole o CDR popisujem samotné využitie a formát záznamu v akej som ju spracovával.

3.1 Záznam podrobností o hovore

Záznam podrobností o hovore (Call detail records) uchováva informácie o hovoroch uskutočnených prostredníctvom telefónnej služby. Záznam poskytuje presné odpovede na otázku kto, kde, kedy a ako dlho uskutočnil hovor.

```
/* příchozí
210220114159+525+  xxxx+000005+00420603xxxxxx        +00000+
      a       b     c      d          e                f       g

/* odchozí
$210220114633+471+  xxxx+000016+739xxxxxx             +00002+
      a        b     c      d        e                  f       g
```

Obr. 3.1: Call detail record časť 1.

```
/* příchozí
+  +00000089+      +00+05+2+0283+07+00+0+1+01+ 507623+{CR}{LF}
  h     i       j    k  l m   n   o  p q r  s     t

/* odchozí
 +00+00000066+      +00+03+3+0178+01+00+0+1+01+ 507654+{CR}{LF}
   h     i       j    k  l m   n   o  p q r  s     t
```

Obr. 3.2: Call detail record časť 2.

Ide zároveň ale aj o cenný nástroj pre firmy s účelom zistenia najvyťaženejšej linky, najvyťaženejšieho zamestnanca či zistenie najvyťaženejšej pracovnej hodiny [16]. Tento druh záznamu tak umožňuje súhrnne sledovať volaciu aktivitu a konkrétne obsahuje metadáta o tom, aké číslo využíva telefónny systém. Obrázky 3.1 a 3.2 boli oddelené pre lepšiu čitateľnosť, inak figurujú súčasne.

Obrázok 3.1 a 3.2 zobrazujú reálne uskutočnené hovory premietnuté do databázy CDR pre oba prípady a teda hovor prichádzajúci a hovor odchádzajúci. Údaje sú oddelené znakom "+". Vysvetlené sú v poradí v akom sa vyskytujú. Podrobnosti sú nasledovné:

(a) čas začiatku hovoru vo formáte yy/MM/dd/hh/mm/ss kde
   i. yy - rok,
   ii. MM- mesiac,
   iii. dd - deň,
   iv. hh - hodina,
   v. mm - minúta,
   vi. ss - sekunda.

(b) identifikátor kanálu - kanál po ktorom volanie odišlo alebo prišlo

(c) pobočka
   i. reálna - obsahuje štvormiestne číslo
   ii. fiktívna - obsahuje šesťmiestme číslo

(d) dĺžka hovoru vo formáte hh/mm/ss kde
   i. hh - hodina,
   ii. mm - minúta,
   iii. ss - sekunda

(e) číslo z verejnej siete

(f) tarifné pulzy

(g) PIN umožňujúci identifikáciu účastníka

(h) smerový kód

(i) číslo oblasti kde
   i. prvé tri čísla znamenajú číslo oblasti pobočky
   ii. nasledujúce tri čísla sú čísla podoblasti
   iii. posledné dve čísla značia číslo uzlu

(j) číslo predávajúceho účastníka - využívané pri presmerovaní hovoru

(k) trieda PINu

(l) sieť operátora

(m) typ pripojenie

   i. 2 - prichádzajúci hovor

   ii. 3 - odchádzajúci hovor

(n) dĺžka vyzváňania vo formáte 0.1s

(o) doplnkové služby

(p) počet tranzitov

(q) prenosová rýchlosť

(r) textové služby

(s) počet b-kanálov

(t) identifikátor udalosti

3.2 Hašovacia funkcia

Hašovacia funkcia je zobrazenie, ktorému sa priradzuje ľubovoľne dlhá vstupná postupnosť a to podľa predom definovaného algoritmu. Výstupná postupnosť je vždy pevnej dĺžky. Práve táto vlastnosť bola kľúčová pri tvorbe samotnej funkcie, keďže bolo potrebné vyriešiť výpočetnú náročnosť faktorizačného systému RSA vtedy ako najrozšírenejšieho asymetrického systému. Zaistila sa tým konštantná dĺžka doby podpisovania akejkoľvek dlhej správy. Hašovacie funkcie sú jednocestné, čo znamená, že výpočet hašu zo správy je výpočetne výrazne jednoduchší ako spätná rekonštrukcia správy zo samotnej znalosti hašu. Je to určené rozdielom vo veľkosti hašu a možnej veľkosti správy. Funkcia dokáže overiť zhodnosť či rozdielnosť veľkostne objemných súborov. Vypočítaním hodnôt ich hašov a vzájomným porovnaním je možné zistiť, či sú haše zhodné. Potom platí, že sú zhodné aj vstupné správy z ktorých bol haš vypočítaný[17].

3.3 Hašovacia tabuľka

Dátová štruktúra združujúca hašovací kľúč so zodpovedajúcimi dátami sa nazýva hašovacia tabuľka. Hašovací kľúč je výsledkom hašovacej funkcie a je využívaný k rýchlemu vyhľadávaniu položiek v poli alebo iného homogenného dátového typu. Homogénny dátový typ je taký, v ktorom sa nachádzajú všetky položky podobného typu[18]. Hašovacou funkciou priradzujeme hodnotu indexu kľúča do homogénnej dátovej štruktúry. Pri zápise obsahu položky sa zapíše na zodpovedajúce miesto. V prípade, že je miesto obsadené, sa pomocou vhodného algoritmu priradí ďalší index nasledujúci v poradí. Vygenerovaním položky sa spočíta pomocou kľúča index hľadanej položky. Pokiaľ bolo dané miesto prepísané položkou s iným kľúčom, opäť sa využije vhodný algoritmus na prehľadanie ďalšej položky. Zvolením správnej veľkosti homogénnej dátovej štruktúry a zvolením hašovacej funkcie má tento algoritmus konštantnú zložitosť.

3.4 Bezpečnostné hash algoritmy

Zatiaľ čo šifry chránia dôvernosť dát v snahe zaručiť, že dáta odoslané v čistom formáte nie je možné prečítať, haš funkcia chráni integritu dát v snahe zaručiť, že dáta neboli počas prenosu upravované. Platí podmienka, že ak je funkcia haš zabezpečená, potom by dve odlišné vety mali mať vždy iný haš. Pokiaľ sa v správe zmení čo i len jeden bit, bude výsledná haš správa úplne odlišná[19]. Všetky hašovacie funkcie vyvinuté od 80. rokov až do roku 2010 sú založené na konštrukcii Merkle - Damgard: MD4, MD5, SHA-1 a rodina SHA-2. Táto konštrukcia nie je dokonalá, avšak je jednoduchá, preukázateľne bezpečná pri novších implementáciach a pre následné využitie v aplikáciach.

3.4.1 MD5

MD5 je hašovacou funkciou so 128 bitovou výstupnou dĺžkou. Funkcia bola navrhnutá v roku 1991 a určitú dobu sa verilo, že je rezistentná voči kolíziam. Po niekoľkých rokoch sa v MD5 začali objavovať rôzne zraniteľnosti ale tie ešte neviedli k nájdeniu kolízie. Až roku 2004 predstavil čínsky tím kryptoanalytikov novú metódu prelomenia šifry a preukázal explicitnú kolíziu. Tento útok bol časom vylepšený a dnes je možné nájsť kolízie v MD5 za menej než minútu.

3.4.2 SHA-1

Hašovacia funkcia SHA-1, štandardizovaná v roku 1995 má 160 bitovú výstupnú dĺžku. V roku 2005 teoretická analýza vykázala možnosť kolízie v šifre a je možné dosiahnuť zhruba 2^{69} vyhodnotenia hašu, čo je omnoho menej než 2^{80} v tzv. narodeninovom útoku. Odvtedy sa šifra neodporúča využívať. V roku 2017 sa dosiahla explicitná zrážka, pričom útok vyžadoval ekvivalent 2^{63} haš vyhodnotení[20].

3.4.3 SHA-2

Hašovacia rodina funkcie SHA-2 bola predstavená v roku 2001 a skladá sa z dvoch súvisiacich hash funkcií SHA-256 a SHA-512 s 256 bitovou respektíve s 512 bitovou výstupnou dĺžkou. Výstupné dĺžky môžu byť skrátené. Momentálne túto hašovaciu funkciu nepostihujú podobné slabosti, ktoré viedli k útokom na SHA-1. Taktiež vďaka dĺžke samotnej haše bude i obtiažnejšie nájsť možnú kolíziu. V súčastnosti je táto hašovacia funkcia stále odporúčaná na použitie i keď už existujú malé zraniteľnosti.

Táto funkcia bola využitá v práci v súvislosti s potrebou anonymizovania citlivých údajov, ktoré nebolo možné ponechať v pôvodnej podobe.

3.4.4 SHA-3

Vzhľadom na kolízny útok na MD5 a teoretické nedostatky na funkcii SHA-1 bola vyhlásená v roku 2007 súťaž na návrh novej kryptografickej hašovacej funkcie. V roku 2012 z celkových 51 kandidátov vzišiel víťazný návrh od spoločnosti Keccak ktorý je označovaný ako SHA-3 s podporou výstupnej dĺžky 224, 256, 384 a 512 bitov. Štruktúra samotnej funkcie je výrazne pozmenená voči SHA-1 a SHA-2. Nepoužíva transformáciu predchodcov Merkle - Damgard čo bol aj jeden z dôvodov, prečo bola vybratá. Samotné jadro Keccak je tvorené nekľúčovou permutáciou s veľkosťou bloku 1600 bitov. Hašovacia funkcia je odolná voči kolíziam za predpokladu, že permutácia je volená ako náhodná.

3.5 Pobočková ústredňa OpenScape 4000

Ústredňa použitá na VŠB-TUO sa volá OpenScape 4000 od spoločnosti Unify. Je určená pre stredné a veľké firmy až do 12 000 účastníkov. Práve táto ústredňa je zdrojom CDR dát, ktoré som použil na štatistické spracovanie v mojej práci. Medzi vlastnosti ústredne patrí podpora volania skrz klasické telefónne služby ale aj VOIP, implementácia šifrovania prenosu hlasu skrz VOIP, pripojenie cez protokol SIP na verejného operátora a iné [21].

Kapitola 4

Návrh a implementácia anonymizácie datovej sady

4.1 Implementácia

Cieľom mojej práce bolo vytvoriť program, ktorý dokáže získať informácie z rozsiahlej sady dát obsahujúcich CDR záznamy za zvolené obdobie. Pre implementáciu som sa rozhodol využiť programovací jazyk Python, ktorý je taktiež označovaný ako skriptovací jazyk pre web aplikácie. Výber Pythonu bol z dôvodu potrebnej implementácie matematických modelov a výslednej kresbe grafov, ktorý je vďaka množstve knižníc jazyka jednoduchšie umožniteľný. Program je založený na GUI prostredí Pythonu zvané ako Tkinter. Vývojárske prostredie som zvolil PyCharm a následne pri spracovaní väčšieho množstva dát som musel migrovať na prostredie Anaconda, pretože PyCharm mal problémy s využitím plnej operačnej pamäte aj napriek tomu, že bola využitá PyCharm 64-bit verzia. Celkovo sa mi ale viac páčilo pracovať v prostredí PyCharm, ktoré sa mi zdalo prehľadnejšie a malo modernejší vzhľad.

4.1.1 Programovací jazyk Python

Python je široko využívaný, objektovo orientovaný programovací jazyk používaný pre univerzálne programovanie. Jeho zaujímavosťou je, že syntax sa snaží byť jednoducho zrozumiteľná ako obyčajná angličtina [22]. Aj vďaka tejto vlastnosti sa pomaly vyzdvihuje v rebríčku najviac používaných programovacích jazykov, kde momentálne už obsadzuje tretiu pozíciu. Prvá verzia jazyka bola vydaná v roku 1991 ako open-source, takže k jeho rozvoji môže prispieť ktokoľvek.

4.1.2 Tkinter

Jazyk Python využíva niekoľko možností pre vývoj GUI. Zo všetkých metód je najčastejšie používaná metóda tkinter, ktorá je známa ako štandardizované rozhranie Pythonu. Python s tkinter je

najrýchlejší a najjednoduchší spôsob, ako vytvoriť aplikáciu s GUI.

4.1.3 Vývojové prostredie PyCharm

Prostredie, ktoré som využil z väčšej časti pri vývoji programu je vyvinuté českou spoločnosťou JetBrains. Poskytuje analýzu kódu, grafický debugger a integrovaný tester jednotiek. Prostredie je multiplatformné naprieč systémami Windows, macOS či Linux. PyCharm Community Edition ktorú som využil je voľná a prístupná pod Apache Lincense. Výhodou tohto projektu je open source, takže je možné ho ľubovoľne používať a modifikovať podľa potreby [23] .

4.1.4 JSON

JavaScript Object Notation je štandart formátu súboru s ktorým som primárne pracoval pri spracovaní údajov v tejto práci. Jedná sa o veľmi často využívaný dátový formát, ktorý bol odvodený z javascriptu [24]. Používa sa zvyčajne pre dátové objekty skladajúce sa z datových typov poľa a iných hodnôt. Práve jednoduchosť vyťahovania údajov z formátu JSON sa mi zapáčila a preto som zvážil následné využitie JSON na konverziu zo surových dát.

4.2 Konverzia dátovej sady

Dáta, ktoré som dostal k analýze boli vo formáte .bak. Uvedený formát sa využíva na záložné kópie súborov. Keďže dáta neboli vo vhodnom stave na spracovanie, vytvoril som konvertor, kde sa celý súbor prekonvertoval do formátu json. Konverter fungoval na spôsobe oddeľovania premenných na základe znaku "+", ktorým boli dáta v surovej podobe oddeľované a celý riadok sa upravil do json objektu s vlastným prístupovým kódom a číslom. V tomto výstupnom súbore mal každý údaj svoju premennú a príslušnú hodnotu.

```
l = 1  # count variable for call id creation
for line in fh:
    description = list(line.strip().split('+', 20)) # reading line by line
        from the text file, if "+" then split and add to variable
    sno = 'call' + str(l)  # for automatic creation of id for each call
    i = 0  # loop variable
    dict2 = {} # intermediate dictionary
    while i < len(fields):
        # creating dictionary for each call
        dict2[fields[i]] = description[i]
        i = i + 1
    # appending the record of each call to the main dictionary
```

```
        dict1[sno] = dict2
        l = l + 1
```

Listing 4.1: Ukážka využitej konverzie do json v jazyku Python

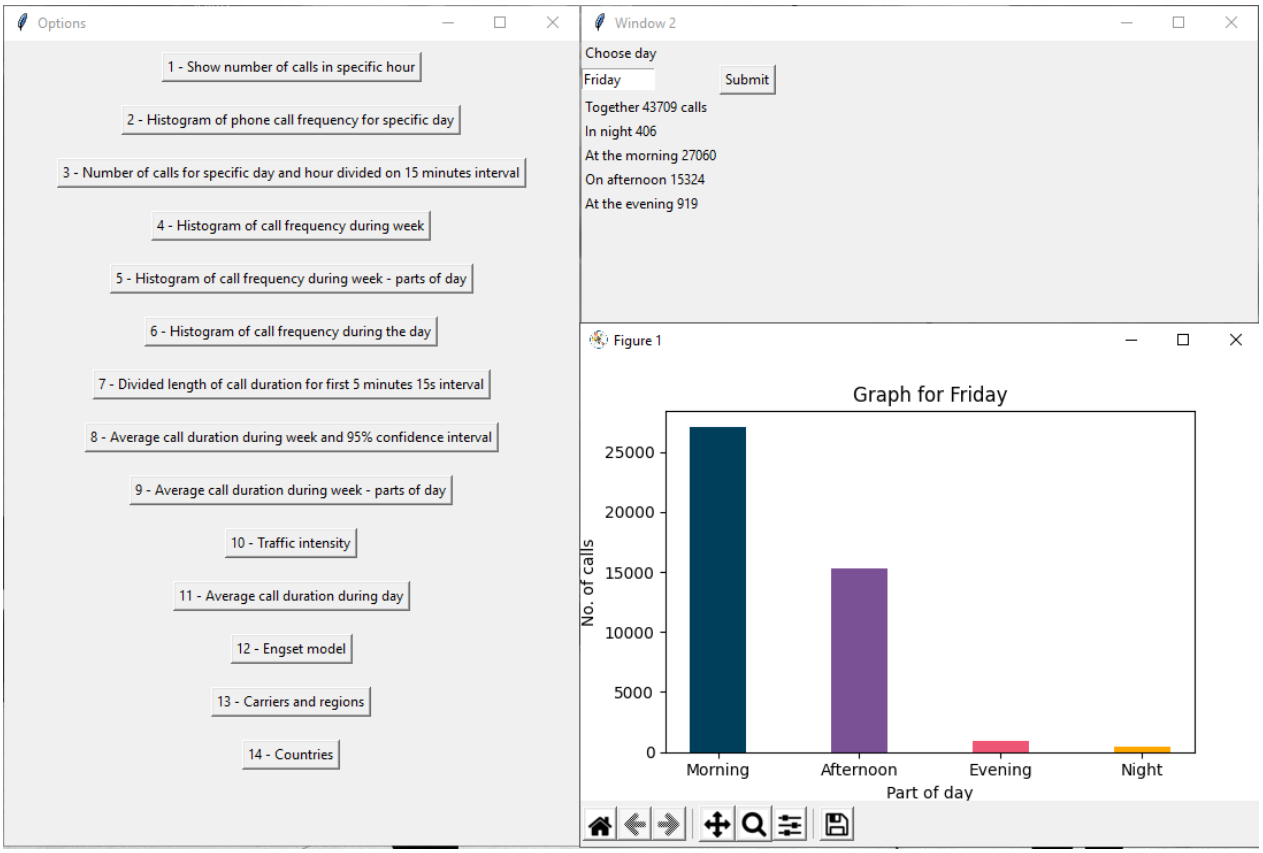
4.2.1 Anonymizácia údajov

Tieto surové dáta obsahovali citlivé údaje ako číslo pobočky a číslo z verejnej siete, ktoré museli byť z dôvodu zachovania súkromia pozmenené a to nasledovne. Čísla z verejnej siete boli skrátené na prvé tri číslice z dôvodu možného vyčítania štatistík o predvoľbe mobilných operátorov. Teda národná predvoľba +420 bola odstránená a zostala len predvoľba operátora. Čísla pobočky boli pozmenené hashovacou funkciou SHA512. Niektoré čísla pobočiek mohli vykazovať repetitívne údaje a teda i hash. Pre vytvorenie unikátnosti riešenia a odstránenia repetitívnych dát bol využitý náhodne sa generujúci UUID identifikátor verzie 4. Vzor údajov podliehajúci úprave je zachytený na obrázku 3.1 a 3.2. Táto konverzia a následná anonymizácia údajov bola najviac časovo a výpočtovo náročná, kedy sa zo surových dát vytvoril json súbor približne osem až deväťnásobne objemnejší voči neupravenej forme vo formáte záložných kópií .bak.

4.3 Grafické rozhranie

Po aplikovaní anonymizácie som mohol priamo pokračovať na implementáciu modelov na vykreslenie grafov a údajov, ktoré som následne aj využil pre štatistické vyhodnotenie. Keďže som mal dataset upravený do formy JSON, samotné vykresľovanie údajov už nebolo náročné. Pomocou podmienok a cyklov som sa dostal k očakávaným údajom. Samotný program má 14 možností práce s dátami. Každá možnosť umožňuje zobraziť analýzu dát pre hovory prichádzajúce a odchádzajúce.

Obrázok 4.1 zobrazuje GUI využité v programe, ovládanie je jednoduché. Uživateľ si zvolí kliknutím funkciu zobrazenia. Následne zvolí požadovaný vstupný parameter, v tomto prípade to bol deň v týždni pre ktorý sa následne zobrazia štatistiky a vyobrazí histogram.



Obr. 4.1: Ukážka grafického prostredia aplikácie

Kapitola 5

Štatistické vyhodnotenie prevádzkových charakteristík

Sledované údaje obsiahnuté v analýzach opisujú správanie používateľov naprieč jednou dekádou a to konkrétne medzi rokmi 2011 až 2020. Možno tak sledovať postupný pokles celkového objemu realizovaných volaní postupom rokov. Táto skutočnosť môže byť spôsobená zmenou komunikácie koncových používateľov a to konkrétne vplyvom sociálnych sietí, emailov, zvýšenou obľúbenosťou využívania kolaboratívnych nástrojov ako Zoom, Webex či iné.

Rozlišujeme taktiež dva druhy hovorov, prichádzajúce a odchádzajúce. Hovory prichádzajúce sú primárne generované študentmi, ktorý volajú na univerzitu a naopak hovory odchádzajúce zvyčajne patria zamestnancom.

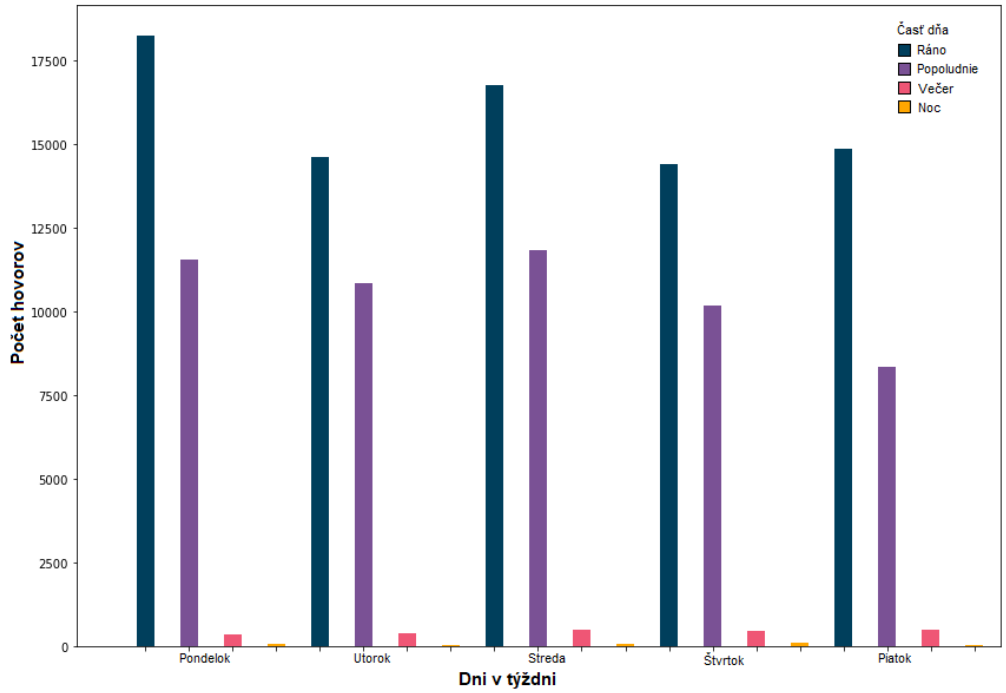
5.1 Počet hovorov

Počet hovorov uskutočnených uživateľmi služby je miera charakterizujúca správanie pri používaní a zároveň indikuje obľúbenosť služby [25]. Celkový počet hovorov uskutočnených počas sledovanej doby desiatich rokov je 4 086 630 hovorov. To v priemere teda činí 1119 hovorov každý deň. Avšak naprieč rokmi môžme sledovať významný pokles. Kým napríklad ešte v roku 2011 bol priemerný počet hovorov za deň 1883 za rok 2020 to už bolo iba 528 hovorov, viď tabuľka 5.1.

Zo štatistiky možno vypozorovať ročný priemerný úbytok počtu volaní o 13%. Najväčší pokles činí medzi rokmi 2016 a 2017 kde možno taktiež vysledovať, že počet prichádzajúcich hovorov začína prevyšovať počet odchádzajúcich. Zaujímavosťou je taktiež pomer 45:55 v počte prichádzajúcich hovorov k počtu odchádzajúcich na začiatku sledovaného obdobia, kedy sa postupom rokov tento pomer obrátil v prospech prichádzajúcich hovorov.

Tabuľka 5.1: Vývoj počtu hovorov naprieč rokmi

Rok	Počet hovorov	Počet prichádzajúcich hovorov	Počet odchádzajúcich hovorov
2011	687 411	298 379	389 032
2012	634 954	273 781	363 173
2013	560 588	246 809	313 779
2014	472 820	217 762	255 058
2015	411 661	195 888	215 773
2016	347 411	168 564	178 847
2017	291 650	149 351	142 299
2018	257 137	136 498	120 639
2019	229 744	126 880	102 864
2020	193 254	106 130	87 124



Obr. 5.1: Histogram počtu hovorov v závislosti od dňa v týždni a časti dňa

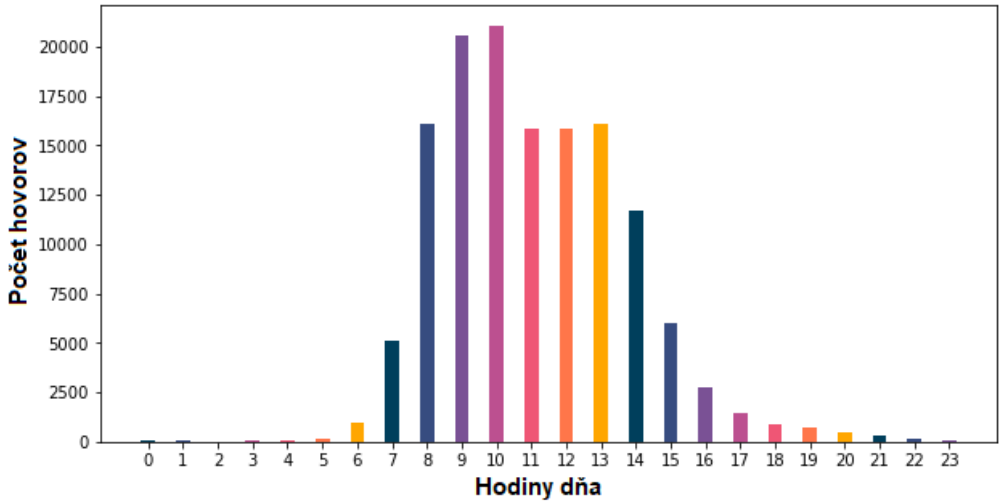
5.1.1 Vplyv dňa v týždni na počet hovorov

Obrázok 5.1 zobrazuje celkový počet všetkých prichádzajúcich hovorov uskutočnených naprieč rokom 2018 používateľmi služby rozdelených podľa dní v týždni. Paradigma používateľov počas celého sledovaného obdobia je podobné a priložené v prílohe. Počas pracovných dní sa počet hovorov pohybuje od 30 147 (pre hovory v pondelok) až po 23 718 (pre hovory v piatok). Hovory uskutočnené cez víkend boli štatisticky zanedbateľné nakoľko ide o firemnú štatistiku tak výrazne prevažuje zaťaženosť počas pracovných dní.

5.1.2 Vplyv dennej doby na počet hovorov

Podrobnejšiu analýzu využitia služby je možné vykonať na rozdelení hovorov podľa času dňa 5.1. Pre analýzu bol deň rozdelený do štyroch periód po 6 hodinách a to ráno od 6:00 do 12:00, popoludní od 12:00 do 18:00, večer od 18:00 do 00:00 a noc od 00:00 do 6:00. Obrázok 5.1 ukazuje významné rozdiely v činnosti medzi rôznym časovým rozpätím. Ráno je najviac vyťažené obdobie na uskutočnenie hlasových hovorov s celkovo 78 807 počtom hovorov (alebo 58,8%). Druhé najvyťaženejšie obdobie je popoludnie spolu s 52 705 počtom hovorov (alebo 39,4%) po ktorom následuje večer spolu s 2 105 hovormi (alebo 1,6%). Nie je prekvapujúce, že najmenej hovorov prebieha v noci a to len 277 (alebo 0,2%).

5.1.3 Vplyv hodiny dňa na počet hovorov



Obr. 5.2: Histogram počtu hovorov v závislosti hodine dňa

Obrázok 5.2 zobrazuje celkový počet všetkých prichádzajúcich hovorov uskutočnených naprieč rokom 2018 používateľmi služby rozdelených podľa hodiny v dni. Jeden stĺpec vyjadruje počet hovorov pre danú hodinu. Teda napríklad stĺpec 8 zobrazuje rozpätie od 8:00 do 8:59. Možné je vidieť typické pracovné zaťaženie. Počas noci (0:00 - 5:00) je aktivita veľmi nízka. Prvé príchody do práce začínajúce ráno (6:00 - 8:00) sa začínajú prejavovať mierne vyšším počtom hlasových hovorov. Ďalej dosahujú vrchol (9:00 - 11:00) až 41 649 hovorov(alebo 30,5%) z celkového počtu. Potom okolo obeda je stabilný počet hlasových hovorov (11:00 - 14:00) a následuje stagnácia a pokles počtu hovorov zapríčinený koncom 8 hodinového pracovného času (14:00 - 18:00). Najmenej vyťažená hodina je v noci (2:00 - 3:00) s 19 hovormi. Pozoruhodný je vzor správania počas špičky kedy pri bližšom skúmaní je možné zistiť najväčšiu frekvenciu uskutočnených hovorov medzi 9:30 až

10:30. Tento vypozorovaný údaj môže byť ďalej využitý na predpoveď zaťaženia systému s ohľadom na rezervovanie sieťových prostriedkov.

5.1.4 Rozdelenie počtu hovorov podľa terminácie v sieti

Spracované údaje CDR umožňujú taktiež vysledovať predvoľbu volajúceho/volaného a teda aj, od akého operátora telefonát smeruje. Do štatistiky som zarátal najväčších mobilných operátorov podľa informáciách o predvoľbách [26]. Zobrazené údaje v tabuľke sú pre vzorový rok 2018 pre hovory prichádzajúce.

Tabuľka 5.2: Rozdelenie počtu hovorov

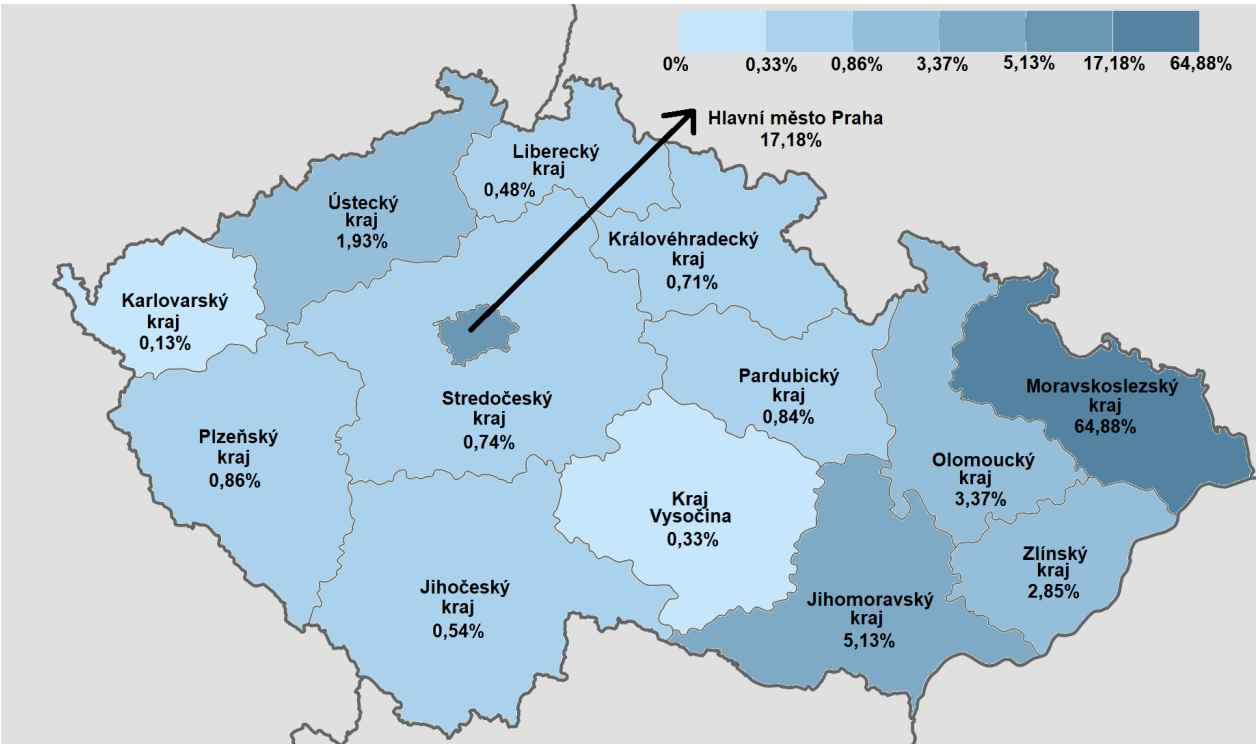
Terminácia do siete	Počet hovorov	Percentuálne vyjadrenie
O2	30 615	25,69%
Vodafone	22 105	18,55%
T-Mobile	38 955	32,69%
Fixná sieť	27 498	23,07%
Spolu	119 174	100%

Z vyobrazených údajov je jasné, že mobilný operátor T-Mobile má jasnú prevahu čo pretrváva naprieč celým sledovaným obdobím 10 rokov. Avšak keď porovnáme bližšie údaje vyčítané naprieč 10 rokov môžme vypozorovať, že udávaný percentuálny pomer sa príliš nemení a dosahuje nanajvýš 2% rozdiel. Problémom zistených údajov však môže byť možný prenos mobilného čísla k inému mobilnému operátorovi kedy je stále viditeľné číslo pôvodné a teda sa do štatistiky zarátava k štatistike operátora od ktorého mobilné číslo pôvodne pochádza.

5.1.5 Rozdelenie počtu hovorov podľa samosprávnych krajov

Údaje z predvolieb nám umožňujú taktiež ukázať pomer volaní do jednotlivých samosprávnych krajov odkiaľ hovor pochádza, avšak len ak ide o hovor z fixnej siete. Vyobrazené údaje sú pre vzorový rok 2018.

Na obrázku 5.3 môžme vidieť percentuálne vyjadrenie zachytených prichádzajúcich volaní z datasetu pre vzorový rok 2018. Prevláda pochopiteľne zastúpenie Moravskoslezského kraja s počtom 17 841 hovorov (alebo 64,88%). Ako druhé je hlavní město Praha s počtom volaní 4 725 (alebo 17,18 %). Najmenší počet hovorov smeroval do Karlovarského kraja a to 36. Ak sa pozrieme na roky ostatné, percentuálne to je veľmi podobné. Výchylka je možná vidieť iba pre hlavní město Praha, odkiaľ v roku 2011 smerovalo 8,5% hovorov. Napriek tomu, že sa jedná výhradne o údaje z fixnej siete, dá sa predpokladať, že podobné údaje by sa mohli vyskytovať i pre údaje pochádzajúce z mobilných telefónov.



Obr. 5.3: Mapa Českej republiky s pomermi volaní do jednotlivých samosprávnych krajov [27]

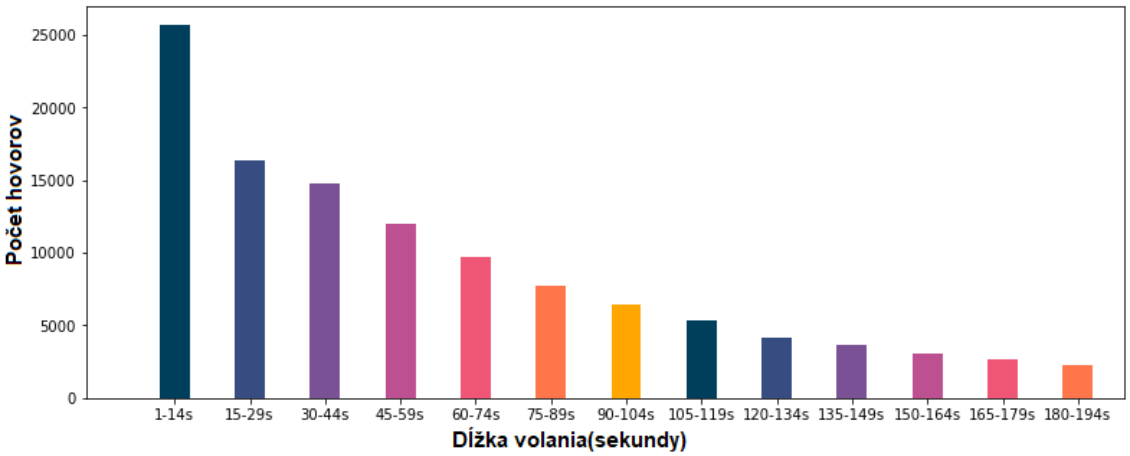
5.2 Dĺžka hovoru

Podobne ako počet hovorov je zaujímavou charakteristikou správania používateľov dĺžka hovoru. Dĺžku hovoru meriame ako časový úsek od doby kedy osoba hovor príjme do okamihu kedy jedna z osôb hovor ukončí. Hovory, ktoré nie sú prijaté a teda majú ako výsledok dĺžku volania 0 sekúnd nie sú zahrnuté do štatistík. Takýchto volaní, ktoré neprebehli bolo pre rok 2018 zaznamenaných 99 776 to znamená 28% z celkového počtu hovorov.

5.2.1 Charakteristika dĺžky hovoru

Priemerná dĺžka hovoru počas hodnotenej doby za rok 2018 je odlišná podľa typu hovoru. Pre hovory prichádzajúce je priemerná dĺžka hovoru 129 sekúnd alebo niečo vyše 2 minút. Avšak pre hovory odchádzajúce je to viac a to až 166 sekúnd a teda takmer 3 minúty.

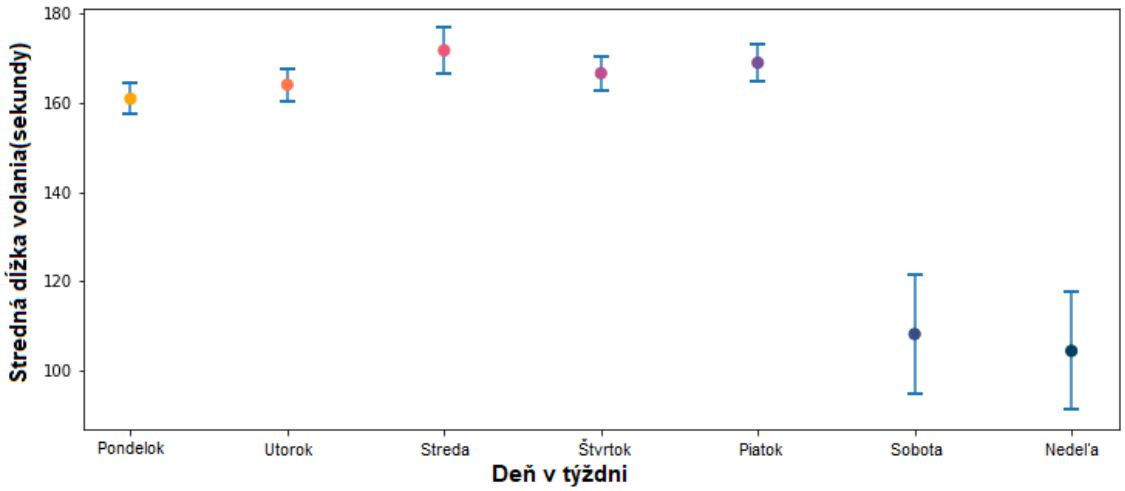
Obrázok 5.4 zobrazuje dĺžku hovoru v intervale 15 sekúnd pre hovory prichádzajúce pre vzorový rok 2018. Ako je znázornené, najviac hovorov prebieha veľmi krátku dobu (kratšie než jednu minútu). Ako sa doba hovoru predlžuje tým sa aj počet hovorov v danom intervale zmenšuje. Špecifickejšie povedané, počet hovorov sa mení podľa Paretovho rozdelenia doby trvania hovoru. Paretova distribúcia je rozdelenie pravdepodobnosti mocenského zákona, ktorý sa zhoduje so spoločenskými, vedeckými, geofyzikálnymi, poistno-matematickými pozorovateľnými javmi [28].



Obr. 5.4: Histogram dĺžky volania

5.2.2 Vplyv dňa v týždni na dĺžku hovoru

Obrázok 5.5 zobrazuje priemernú dobu trvania hlasových hovorov a taktiež 95% interval spoľahlivosti priemernej doby trvania v závislosti na dni v týždni. Interval spoľahlivosti je rozsah hodnôt, u ktorých si môžeme byť istý, že na 95% obsahuje skutočný priemer hodnôt. Toto číslo sa môže líšiť vzhľadom na zvolený interval. Interval taktiež závisí na dátach, ktoré sú zhromaždené. Ak je zvolený interval 99% bude tento interval širší oproti intervalu 90% , pretože bude zobrazovať väčšie množstvo skutočných parametrov.

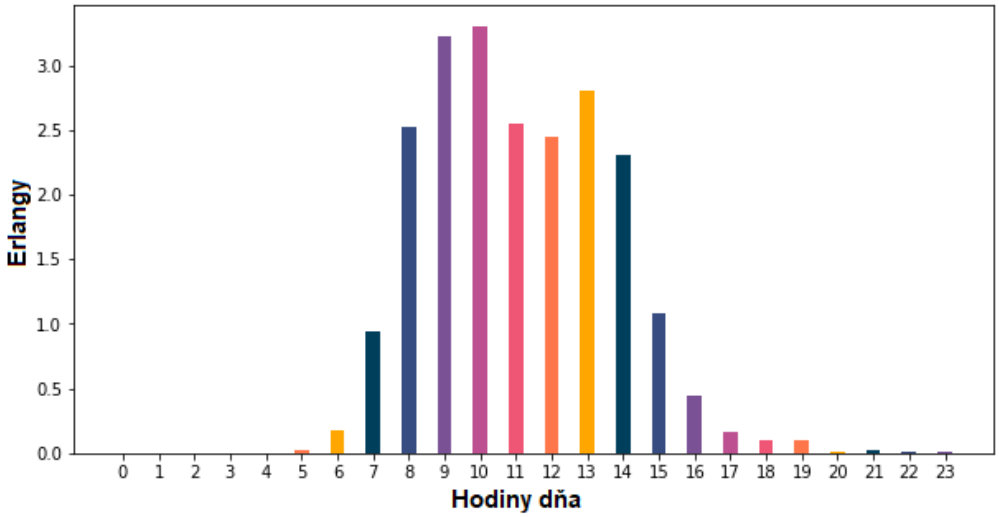


Obr. 5.5: Priemerná doba volania v závislosti na dni v týždni

Histogram na obrázku 5.5 zobrazuje dáta za vzorový rok 2018 pre hovory odchádzajúce. Sú viditeľné rozdiely v priemernej dĺžke hovoru. Pozoruhodná je priemerná doba hovoru v stredu (171,74

s) ktorá je výrazne vyššia ako cez víkendové dni napr. v nedeľu (104,36 s). To je zaujímavé správanie vzhľadom k tomu, že ľudia majú cez víkend viacej voľného času a typicky menej stresujúci deň. Taktiež v pondelok, kedy je najvyšší počet hovorov je doba volania kratšia, čo môže spôsobovať napríklad väčšie pracovné zaťaženie na začiatku týždňa. Ak sa pozrieme bližšie na porovnanie priemernej dĺžky hovoru počas desiatich rokov sa údaje príliš nemenia cez pracovný deň. Avšak cez víkend mimo pracovných dní je možné sledovať významný pokles dĺžky volania kedy v roku 2011 bola priemerná doba hovoru pre nedeľu 217 sekúnd zatiaľ čo v roku 2020 to bolo už iba 62 sekúnd.

5.3 Zaťaženie



Obr. 5.6: Histogram zaťaženia

Zaťaženie udáva obsadenie kanálu behom určitého časového okamžiku, zvyčajne najrušnejšej hodiny. Obrázok 5.6 zobrazuje zaťaženie pre hovory prichádzajúce v období letného semestra pre šk. rok 2017/2018, konkrétne ide o obdobie od začiatku februára až po koniec júna 2018. Jeden stĺpec zobrazuje časový úsek jednej hodiny. Zaťaženie je demonštrované na pondelku, kedy dochádza k najväčšiemu počtu hovorov v celom týždni. Veľkonočný pondelok - sviatok bol z údajov odstránený pre možné skreslenie údajov. Najvyššiu záťaž možno vysledovať medzi desiatou až jedenástou hodinou kedy sa uskutočnilo za sledované obdobie v priemere 100 prichádzajúcich hovorov a pri priemernej dĺžke hovoru 118 sekúnd potom môžme zistiť zaťaženie nasledovne :

$$A = \lambda \cdot T, \tag{5.1}$$

kde A = zaťaženie, λ = počet prichádzajúcich hovorov a T = priemerná dĺžka hovoru. Potom: $\mathrm{A} = \frac{100 \cdot 118}{3600} = 3,28$ Erlang/hod pre úsek najvyťaženejšej hodiny.

Keďže zaťaženie už poznáme, môžme ho aplikovať v rámci formuly Erlang B.

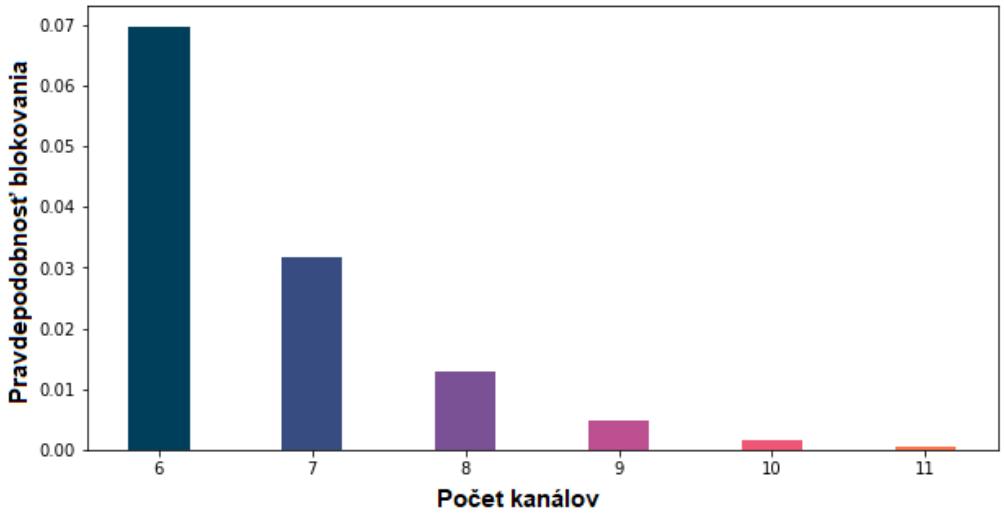
5.3.1 Ukážka modelu Erlang B

Prepokladajme, že máme počet obslužných kanálov $n = 7$ pričom zaťaženie už poznáme $A = 3{,}28$ Erlang. Potom môžme vypočítať s akou pravdepodobnosťou budú všetky kanály obsadené pomocou modelu Erlang B (2.32).

$$P_N = E(7; 3, 28) = \frac{\frac{3{,}28^7}{7!}}{1 + \frac{3{,}28^1}{1!} + \frac{3{,}28^2}{2!} + ... + \frac{3{,}28^6}{6!} + \frac{3{,}28^7}{7!}} = 3, 11\% \qquad (5.2)$$

5.3.2 Ukážka modelu Engset

Na kalkuláciu Engset modelu som využil balíček s názvom fast-engset určený pre programovací jazyk Python [15]. Tento balíček umožňuje priamo zadať parametre a program dopočíta pravdepodobnosť, s akou budú všetky kanály obsadené. Údaje použité na demonštráciu Engset modelu boli identické so zaťažením (5.3) aby bolo možné priamo porovnať uvedené modely. Využitý je dataset pre letný semester akademického roku 2017/2018 pre pondelky v časovom úseku od 10:00 do 11:00. Jedná sa o hovory prichádzajúce.



Obr. 5.7: Pravdepodobnosť blokovania podľa modelu Engset

Z histogramu môžme vyčítať, že pravdepodobnosť, s akou bude hovor blokovaný pri použití šiestich kanálov je $0.0697 \approx 6.97\%$. To je približne rovnaká pravdepodobnosť blokovania aká vyšla z modelu Erlang B. Pre porovnanie tabuľka oboch modelov.

Tabuľka 5.3: Porovnanie pravdepodobnosti blokovania pre modely Erlang B a Engset

Počet kanálov	Erlang B	Engset
5	13,45%	13,63%
6	6,84%	6,97%
7	3,11%	3,18%
8	1,26%	1,29%
9	0,46%	0,47%
10	0,15%	0,15%

Pri použití oboch modelov som dosiahol veľmi zhodné výsledky. Kalkuláciou nižšieho počtu kanálov sa oba modely značne líšia, avšak pri pravdepodobnosti blokovania blížiacej sa nule sa rozdiel zmenšuje. Z výsledkov sa dá zhodnotiť, že ak by bolo potrebné vytvoriť pobočkovú telefónnu ústredňu na uvedené zaťaženie 3.28 Erlang, pre hovory prichádzajúce za predpokladu aspoň 99% priepustnosti by bolo potrebných aspoň 9 obslužných kanálov.

Kapitola 6

Zhodnotenie dosiahnutých výsledkov

Táto práca sa zaoberá analýzou datasetu volaní realizovaných behom desiatich rokov na pobočkovej telefónnej ústredni VŠB-TUO. Implementácia dbá na ochranu osobných údajov použitím bezpečnostného haš algoritmu a náhodne generujúcim sa identifikátorom. Následná konverzia datasetu do formy JSON umožňuje pohodlné spracovanie dát pomocou aplikácie. Aplikácia generuje grafy a výstupy v grafickom prostredí, ktoré sú využité na štatistické zhodnotenie datasetu.

Zaujímavé pozorovanie možno sledovať ohľadom počtu uskutočnených hovorov, kde sledujeme každoročný pokles celkového počtu hovorov v priemere o 13%. Tento pokles môže spôsobovať postupná zmena komunikácie smerom k internetovým službám typu sociálnych sietí či využívanie kolaboratívnych nástrojov. Počas pracovného týždňa prebieha najviac volaní ráno a to konkrétne v rozmedzí pondelka medzi 9:30 až 10:30. Taktiež vplyv dennej doby výrazne vplýva na celkový počet uskutočnených volaní, počas pracovného týždňa viac ako 98% celkového počtu hovorov možno sledovať pre ráno a popoludnie, teda hlavnú pracovnú dobu. Z mobilných operátorov počas celého sledovaného obdobia desiatich rokov prevažujú telefonáty od operátora T-Mobile. Ak sa pozrieme na rozdelenie počtu hovorov podľa samosprávnych krajov vo fixnej sieti, najviac ich smeruje z Moravskoslezského kraja a potom nasleduje Hlavní město Praha. Najdlhšie uskutočnené hovory sledujeme v stredu avšak počas pracovného týždňa nie je príliš výrazný rozdiel v priemernej dĺžke hovorov, naproti tomu cez víkend je razantný prepad dĺžky hovoru.

V závere predvedením modelu Erlang B a Engset dosahujem analogické výsledky pre oba modely kde sa dochádza k zisteniu, že na potrebné zaťaženie pre rok 2018 je potreba aspoň 9 obslužných kanálov pre prichádzajúci smer hovorov.

Literatura

1. SZTRIK, János. *Basic Queueing Theory*. Debrecen, 2012. Dostupné tiež z: `https://irh.inf.unideb.hu/~jsztrik/education/16/SOR_Main_Angol.pdf`.
2. MIŠUTH, T.; BAROŇÁK, I. Application of Erlang B model in modern VoIP networks. In: *2011 34th International Conference on Telecommunications and Signal Processing (TSP)*. 2011, s. 235–239. Dostupné z DOI: `10.1109/TSP.2011.6043736`.
3. GROSS, Donald; SHORTLE, John F; THOMPSON, James M; HARRIS, Carl M. *Fundamentals of queueing theory*. Fifth edition. Hoboken (New Jersey): Wiley, 2017. ISBN 9781118943564.
4. BOLCH, Gunter; GREINER, Stefan; MEER, Hermann de; TRIVEDI, Kishor S. *Queueing Networks and Markov Chains Modeling and Performance Evaluation with Computer Science Applications*. Second Edition. Hoboken (New Jersey): Wiley, 2006. ISBN 9780471565253.
5. DUDIN, Alexander N.; KLIMENOK, Valentina I.; VISHNEVSKY, Vladimir M. *The Theory of Queuing Systems with Correlated Flows*. Cham, Switzerland: Springer, 2020. ISBN 978-3-030-32071-3.
6. TRIVEDI, Kishor S. *Probability and Statistics with Reliability, Queuing and Computer Science Applications*. Second Edition. Duke University, Durham, North Carolina: Wiley, 2016. ISBN 978-0-471-33341-8.
7. MIR, Nader F. *Computer and Communication Networks*. 1st edition. Upper Saddle River (New Jersey): Prentice Hall, 2007. ISBN 0131389106. Dostupné tiež z: `https://flylib.com/books/en/2.959.1.96/1/`.
8. SIGMAN, Karl. *Stochastic Modeling Course*. Columbia University. 2009. Dostupné tiež z: `http://www.columbia.edu/~ks20/stochastic-I/stochastic-I-MCI.pdf`.
9. DOUC, Randal; MOULINES, Eric; PRIOURET, Pierre; SOULIER, Philippe. *Markov Chains*. Springer, 2018. ISBN 978-3-319-97704-1.
10. POLEC, Jaroslav; KARLUBÍKOVÁ, Tatiana; VARGIC, Radoslav. *Pravdepodobnostné modely v telekomunikáciách*. 1. vydanie. Bratislava: Slovenská technická univerzita v Bratislave, 2007. ISBN 978-80-227-2641-2.

11. MIŠUTH, T.; CHROMÝ, E.; KAVACKÝ, M. Prediction of traffic in the contact centers. In: *2009 International Conference on Electrical and Electronics Engineering - ELECO 2009*. 2009, s. II-111-II–114. Dostupné z DOI: 10.1109/ELECO.2009.5355286.

12. VOZŇÁK, Miroslav; MICHÁLEK, Libor. *Sítě nových generací a jejich bezpečnostní problémy pro integrovanou výuku VUT a VŠB-TUO*. 1. vydání. Ostrava: Vysoká škola báňská-Technická univerzita Ostrava, 2014. ISBN 978-80-248-3558-7.

13. CHROMY, E.; DIEZKA, J.; KAVACKY, M.; VOZNAK, M. Markov models and their use for calculations of important traffic parameters of contact center. *WSEAS Transactions on Communications*. 2011, roč. 10, č. 11, s. 341–350.

14. ALZER, H.; KWONG, M.K. On the Erlang loss function. *Acta Mathematica Hungarica*. 2020, roč. 162, č. 1, s. 14–31. Dostupné z DOI: 10.1007/s10474-020-01046-1.

15. AZIMZADEH, P.; CARPENTER, T. Fast Engset computation. *Oper. Res. Lett.* 2016, roč. 44, č. 3, s. 313–318. ISSN 0167-6377. Dostupné z DOI: 10.1016/j.orl.2016.02.011.

16. BARTLEY, Kevin. What Are Call Detail Records (CDRs)? [B. r.], s. 1. Dostupné tiež z: https://www.onsip.com/voip-resources/voip-fundamentals/what-are-call-detail-records-cdrs.

17. HRŮZA, Petr; PITAS, Jaromir; ŠANDA, Jaroslav; BRECHTA, Bohumil. *Kybernetická bezpečnost II*. 2013-01. ISBN 978-80-7231-931-2. Dostupné z DOI: 10.13140/RG.2.1.4193.9047.

18. MAILUND, Thomas. *The Joys of Hashing: Hash Table Programming with C*. Aarhus: Apress, 2019. ISBN 978-1-4842-4065-6.

19. AUMASSON, Jean-Philippe. *Serious Cryptography: A Practical Introduction to Modern Encryption*. No Starch Press, 2018. ISBN 1593278268, ISBN 9781593278267.

20. KATZ, Jonathan; LINDELL, Yehuda. *Introduction to Modern Cryptography*. 3. vyd. CRC Press, 2020. Chapman Hall/CRC Cryptography and Network Security Series. ISBN 0815354363, ISBN 9780815354369.

21. *Unify OpenScape 4000*. [B. r.]. Dostupné tiež z: https://unify.com/en/?nx_doc_id=817df1d0-3d2e-4ff3-983d-c881740bf2ff%5C&type=pdf%5C&action=view.

22. *What is Python?* [B. r.]. Dostupné tiež z: https://pythoninstitute.org/what-is-python/.

23. *Pycharm*. [B. r.]. Dostupné tiež z: https://www.jetbrains.com/pycharm/.

24. *JSON*. [B. r.]. Dostupné tiež z: https://en.wikipedia.org/wiki/JSON.

25. PESSEMIER, Toon De; STEVENS, Isabelle; MAREZ, Lieven De; MARTENS, Luc; JOSEPH, Wout. Quality assessment and usage behavior of a mobile voice-over-IP service. In: Telecommunication Systems, 2015, s. 417–432. Dostupné tiež z: https://link.springer.com/content/pdf/10.1007/s11235-014-9961-9.pdf.

26. *Předvolby síťových mobilních operátorů (nikoli virtuálních operátorů)*. [B. r.]. Dostupné tiež z: https://www.predvolby.cz/inpage/mobilni-operatori/.

27. *Czech Republic location map*. [B. r.]. Dostupné tiež z: https://upload.wikimedia.org/wikipedia/commons/thumb/4/4a/Czech_Republic_location_map.svg/1920px-Czech_Republic_location_map.svg.png.

28. *Pareto distribution*. San Francisco (CA): Wikimedia Foundation, 2001-. Dostupné tiež z: https://en.wikipedia.org/wiki/Pareto_distribution.