
**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**
**BRNO UNIVERSITY OF TECHNOLOGY**

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**
**FACULTY OF INFORMATION TECHNOLOGY**

**ÚSTAV INFORMAČNÍCH SYSTÉMŮ**
**DEPARTMENT OF INFORMATION SYSTEMS**


# ZABEZPEČENÝ VZDÁLENÝ PŘÍSTUP A SPRÁVA POČÍTAČŮ

**SECURE REMOTE COMPUTER MANAGEMENT SYSTEM**

**BAKALÁŘSKÁ PRÁCE**
**BACHELOR'S THESIS**


**AUTOR PRÁCE** **PETER HEŠKO**
**AUTHOR**

**VEDOUCÍ PRÁCE** **Ing. LIBOR POLČÁK, Ph.D.**
**SUPERVISOR**



**BRNO 2019**

**Vysoké učení technické v Brně**
Fakulta informačních technologií

Ústav informačních systémů (UIFS) Akademický rok 2018/2019

# Zadání bakalářské práce


Student: **Heško Peter**
Program: Informační technologie
Název: **Zabezpečený vzdálený přístup a správa počítačů**
**Secure Remote Computer Management System**
Kategorie: Počítačové sítě

Zadání:

1. Nastudujte existující řešení pro vzdálený přístup k Linuxovým systémům včetně grafického rozhraní.
2. Seznamte se s přístupy pro připojení k uzlům bez veřejné adresy z Internetu.
3. Navrhněte software pro zabezpečený vzdálený přístup a správu počítačů bez nutnosti veřejné IP adresy na vzdálených počítačích se systémem Linux dle požadavků firmy Daite s.r.o.
4. Návrh implementujte.
5. Implementaci otestujte a srovnejte s existujícími nástroji.
6. Navrhněte možná vylepšení práce.

Literatura:

- Corbet, J. Virtual private networks with WireGuard. Linux Weekly News, 2018. Dostupné online, URL https://lwn.net/Articles/748582/.

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 až 3 zadání.

Podrobné závazné pokyny pro vypracování práce viz http://www.fit.vutbr.cz/info/szz/

Vedoucí práce: **Polčák Libor, Ing., Ph.D.**
Vedoucí ústavu: Kolář Dušan, doc. Dr. Ing.
Datum zadání: 1. listopadu 2018
Datum odevzdání: 15. května 2019
Datum schválení: 9. května 2019

# Abstrakt

Nasadzovanie a správa vzdialených systémov so sebou prinášajú niekoľko problémov. Ich administratíva a monitorovanie je oveľa rýchlejšie a finančne efektívnejšie, ak je vykonávané na diaľku, ako keď je nutný fyzický prístup k systému. Softvér, ktorý toto umožňuje nie je v mnohých prípadoch voľne dostupný, alebo má obmedzenú funkcionalitu. V tejto práci sa zaoberám analýzou dostupných riešení a následne návrhom a implementáciou aplikácie, ktorá spája všetky požadované vlastnosti. Výsledkom je softvér, pomocou ktorého je možný zabezpečený prístup na vzdialené počítače s operačným systémom Ubuntu 16.04 a ich monitorovanie.



# Abstract

Deployment and management of remote systems poses several problems. Their administration and monitoring is much faster and more cost-effective when done remotely than when physical access to the system is required. Software that allows this is in many cases not freely available or has limited functionality. In this work I deal with the analysis of available solutions and then design and implementation of an application that combines all the required features. The result is software that provides secure access to and monitoring of remote computers running Ubuntu 16.04.



# Kľúčové slová

Vzdialený prístup, monitorovanie, online podpora, SSH, VNC, NAT, VPN, Linux, Ubuntu, x11vnc, Anydesk, TeamViewer, Remina, OpenSSH, OpenVPN

# Keywords

Remote access, monitoring, on-line support, SSH, VNC, NAT, VPN, Linux, Ubuntu, x11vnc, AnyDesk, TeamViewer, Remina, OpenSSH, OpenVPN



# Citácia

HEŠKO, Peter. *Zabezpečený vzdálený přístup a správa počítačů.* Brno, 2019. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Libor Polčák, Ph.D.

# Zabezpečený vzdálený přístup a správa počítačů

## Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením pána Ing. Libora Polčáka, Ph.D. Ďalšie informácie mi poskytli Mgr. Pavel Kohoutek a firma Daite s.r.o. Uviedol som všetky literárne zdroje a publikácie, z ktorých som čerpal.

.....
Peter Heško
31. júla 2019


## Poďakovanie

Chcel by som poďakovať vedúcemu mojej práce, pánovi Ing. Liborovi Polčákovi, Ph.D. za trpezlivosť pri spolupráci a všetky cenné rady, ktoré mi poskytol. Tiež by som chcel touto formou vyjadriť poďakovanie zamestnancom firmy Daite s.r.o., najmä pánovi Mgr. Pavlovi Kohoutkovi za zdieľané skúsenosti a postrehy k práci. Ďakujem aj mojím rodičom, priateľke a kamarátom, ktorí ma podporovali v štúdiu a vždy ma dokázali povzbudiť.

# Obsah

# Kapitola 1

# Úvod

V dobe, keď internet ešte nebol rozšírený tak ako dnes, bolo obtiažne spravovať väčšie množstvo počítačov, najmä ak neboli jednoducho fyzicky dostupné. S narastajúcou dostupnosťou a rýchlosťou internetu sa ale situácia zmenila. Dnes môžeme ovládať a monitorovať prakticky neobmedzené množstvo systémov kdekoľvek na svete bez toho, aby sme museli opustiť svoj počítač. Aby to bolo možné, je nutné splniť dve podmienky. Prvou je, aby malo vzdialené zariadenie prístup k internetu. Druhou podmienkou je špecializovaný softvér, ktorý umožňuje vzdialené ovládanie a monitorovanie. Prvú podmienku väčšinou nie je ťažké splniť, pri druhej ale často môžeme naraziť na rôzne problémy. Existujúce softvérové riešenia totiž veľakrát nie sú podporované na požadovaných platformách, nie sú voľne dostupné, alebo ak sú, majú obmedzenú funkcionalitu. Pri použití voľne dostupných riešení často nastáva problém s neintuitívnym ovládaním a jednotlivé funkcie sú decentralizované do niekoľkých aplikácii. Pritom je pri správe vzdialených počítačov častá pomoc užívateľom, ktorí pri počítači fyzicky sú. Z toho vyplýva potreba interakčnosti takýchto nástrojov. V tejto bakalárskej práci sa v druhej kapitole venujem analýze komerčných a aj voľne dostupných riešení. Cieľom skúmania sú konkrétne programy TeamViewer a Anydesk z kategórie komerčných riešení a na strane voľne dostupných riešení sa zaoberám rôznymi nástrojmi ako napríklad x11vnc, Remmina, OpenSSH a iné. V tretej kapitole skúmam možné technológie, ktoré by dokázali prepojiť dva počítače, aj keď jeden z nich, alebo oba nemaju statickú verejnú IP adresu. V štvrtej kapitole analyzujem skúmané riešenia a technológie potrebné na dosiahnutie požadovanej funkcionality a vyberám z nich tie najlepšie, ktoré neskôr použijem pri implementácii vlastného nástroja. V piatej kapitole je popisaná implementácia oboch častí vytvoreného nástroja Support tool. V šiestej kapitole testujem splnenie požadovanej funkčnosti, spoľahlivosť zabezpečenia a celkovú funkčnosť častí nástroja.

# Kapitola 2

# Analýza existujúcich riešení

Existuje mnoho aplikácií, ktoré umožňujú vzdialený prístup na plochu alebo terminál vzdialených počítačov, alebo ich monitorovanie. Táto kapitola sa zaoberá vybranými aplikáciami, ktoré svojimi vlastnosťami aspoň z časti vyhovujú požiadavkám, ktoré boli navrhnuté v spolupráci s firmou Daite s.r.o. Všetky analyzované aplikácie však majú spoločné, že im chýba niektorá z kľúčových vlastností, vďaka čomu je v súčasnom stave nutné používať kombináciu týchto softvérových nástrojov pre dosiahnutie požadovanej funkcionality. V kapitole sú popísané požiadavky na softvérové riešenie a následne postup používania jednotlivých nástrojov, ako aj podrobný popis výhod a nevýhod daných nástrojov.

## 2.1 Požiadavky

Uvedené požiadavky špecifikujú vlastnosti, aké by mal mať vytvorený softvér. Tiež sú v podstate dôvodom, prečo nie je možné použiť nejaké už existujúce softvérové riešenia. Cieľom určenia požiadaviek je návrh a implementácia prototypu aplikácie, ktorá by pri ich splnení mohla slúžiť ako základ softvéru riešiaceho reálny problém z praxe. Navrhnuté požiadavky sú nasledovné:

**Podpora operačného systému Ubuntu 16.04** Ubuntu 16.04[1] je stabilná verzia s dlhodobou podporou, ktorá v súčasnosti patrí medzi najpopulárnejšie distribúcie Linuxu. V tomto operačnom systéme je mnoho nástrojov na monitorovanie, alebo vzdialený prístup, zahrnutých už v základnej inštalácii. Ďalšie užitočné nástroje sú často voľne dostupné a relatívne jednoducho nainštalovateľné. Zároveň väčšina systémov, ktoré firma Daite s.r.o. spravuje je založená na operačnom systéme Ubuntu 16.04. Z toho vyplýva, že daný nástroj by mal byť schopný spravovať a monitorovať najmä systémy s touto konkrétnou verziou distribúcie Ubuntu. Výhodou, ale nie nutnosťou, je potom kompatibilita aplikácie na správu a monitorovanie s ďalšími verziami Ubuntu, inými Linuxovými distribúciami alebo odlišnými operačnými systémami.

**Funkcionalita** Pre efektivitu práce pri spravovaní, monitorovaní a poskytovaní podpory užívateľom na vzdialenom systéme je dôležité, aby daný nástroj spájal potrebnú funkcionalitu do jedného celku. Je teda žiadúce, aby boli jednotlivé funkcie čo najjednoduchšie využiteľné. Medzi požadované funkcie patrí zabezpečený prístup na terminál vzdialeného

[1] Oficiálna webová stránka distribúcie Ubuntu https://www.ubuntu.com/

počítača, zabezpečený prístup na jeho plochu, zabezpečený prenos súborov, uloženie zoznamu vzdialených systémov, ktoré sú predmetom spravovania a monitorovania a zobrazenie informácií o ich softvérovej a hardvérovej konfigurácii, ako aj aktuálne údaje o zaťažení jednotlivých komponentov. Taktiež je nutná možnosť pripojenia sa aj po reštartovaní vzdialeného počítača.

**Možnosť pripojenia sa na vzdialenú plochu, monitorovania a správy počítačov aj bez nutnosti statickej verejnej IP adresy** Väčšina vzdialených systémov, ktoré sú predmetom spravovania a monitorovania, nie je dostupná cez statickú verejnú IP adresu. Z tohto dôvodu je teda kriticky potrebné, aby výsledný systém dokázal vytvoriť spojenie aj medzi počítačmi, z ktorých jeden, alebo oba budú pripojené v lokálnej sieti a smerovač, cez ktorý budú pripojené na internet, nebude mať statickú verejnú IP adresu.

**Spoľahlivosť, stabilita a bezpečnosť pripojenia** Pri poskytovaní online podpory systémov niekedy vznikajú situácie, kedy je dôležitá každá minúta, z čoho vyplýva, že stabilita a spoľahlivosť pripojenia sú veľmi dôležité. Zároveň sú prenášané veľmi citlivé informácie a neautorizovaný prístup na spravované systémy je striktne zakázaný. Z týchto dôvodov je potrebné komunikáciu adekvátne zabezpečiť.

**Prehľadné a jednoducho čitateľné užívateľské rozhranie** S vyšším množstvom relevantných informácií narastá šanca na rýchlejšie vyriešenie problému, avšak iba v prípade, ak sú dané informácie jednoducho čitateľné. V opačnom prípade má väčšie množstvo informácií skôr negatívny dopad na schopnosť rýchlo odstrániť závadu, alebo poskytnúť podporu na vzdialenom systéme. Softvér by mal zobrazovať informácie o hardvérovej a softvérovej konfigurácii jednotlivých počítačov. Tie by mali byť prehľadne roztriedené do jednotlivých kategórií pre rýchlu orientáciu.

**Cena** Cieľom tejto práce je, okrem iného, aj zredukovať náklady vynaložené na kúpu licencií pre softvérové riešenia tretích strán – ich funkcionalita je niekedy nevyužitá a niekedy nedostačujúca. Využitím voľne šíriteľných nástrojov na správu a monitorovanie vzdialených systémov sa výrazne zvýši efektivita takto vynaložených zdrojov.

## 2.2 TeamViewer

TeamViewer[2] je jedna z najznámejších aplikácií na vzdialený prístup medzi bežnými používateľmi. Prispieva k tomu fakt, že je veľmi jednoduché ho nainštalovať a používať a je tiež multiplatformný. Je teda možné prepojiť aj dva počítače s rôznymi operačnými systémami, alebo aj pripojiť sa na vzdialený počítač s ľubovoľným operačným systémom z mobilného telefónu s operačným systémom Android, alebo iOS. Na platforme Microsoft Windows je dokonca možné TeamViewer používať bez inštalácie ako prenosnú verziu.

### Kompatibilita s operačným systémom Ubuntu

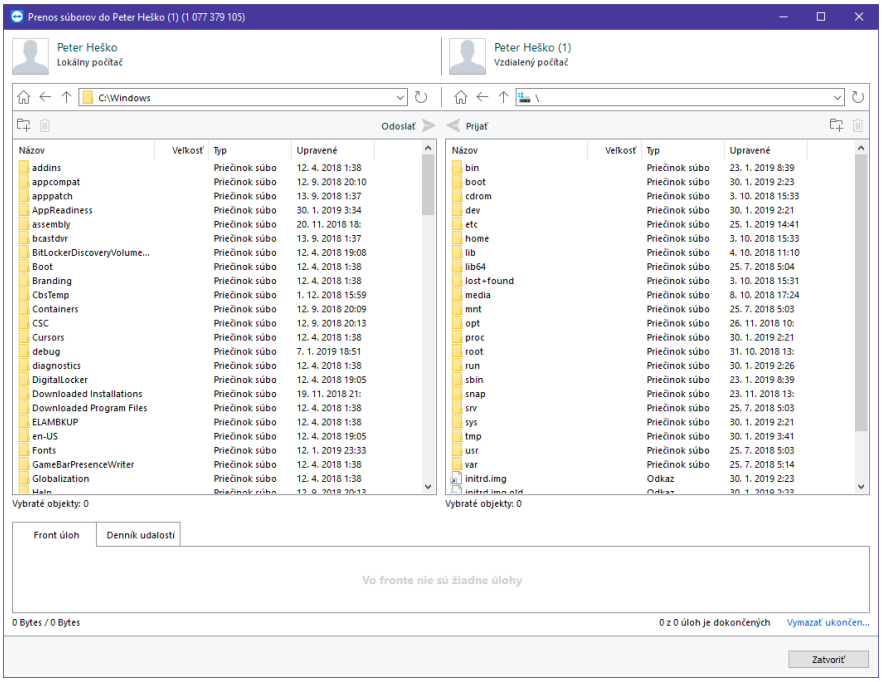
Informácie k tomuto odstavcu som čerpal z oficiálneho komunitného príspevku podpory TeamVieweru a z jeho predchádzajúcich úprav [22]. Najnovšia verzia TeamVieweru (v čase písanie tejto práce je to verzia TeamViewer 14) je oficiálne podporovaná pre operačný

[2]Oficiálna webová stránka aplikácie TeamViewer https://www.teamviewer.com/

systém Ubuntu 16.04 a tiež pre Ubuntu 18.04. Oficiálnu podporu pre Ubuntu 16.04 má aj predchádzajúca verzia 13. Tá oficialne podporuje aj verziu Ubuntu 18.04 a tiež verziu Ubuntu 14.04. Predchádzajúce verzie TeamVieweru (verzia 5 až 7) taktiež podporujú operačný systém Ubuntu. Na oficiálnych stránkach však nie je uvedená konkrétna verzia operačného systému [16]. Pri verziách TeamVieweru 8 až 12 nie je na oficiálnych stránkach uvedená podpora pre operačný systém Ubuntu, je však možné stiahnuť inštalačný balík deb pre 64 aj 32 bitové architektúry [15]. Testoval som TeamViewer verziu 13 a 14 na operačných systémoch Ubuntu 16.04 LTS (Xenial Xerxus) a Ubuntu 18.04 LTS (Bionic Beaver). Obe verzie sú u dvoch vyššie spomínaných systémoch plne funkčné.

### Funkcie

Teamviewer má niekoľko funkcií. Okrem tej hlavnej, ktorou je prístup na vzdialenú plochu, je v najnovšej verzii možné aj prenášať súbory, posielať rýchle správy na vzdialené počítače, nahrávať záznam spojenia, hlasovo konverzovať s užívateľom vzdialeného počítača, usporiadať schôdzky, či reštartovať vzdialený počítač alebo na ňom vyvolať klávesovú skratku Ctrl-Alt-Delete. Množstvo funkcií ale závisí od konkrétnej nainštalovanej verzie softvéru. Taktiež záleží aj od operačného systému, ktorý je nainštalovaný ako na strane počítača, z ktorého sa pripájame, tak aj na strane počítača, na ktorý sa pripájame. V spojení medzi dvomi počítačmi s operačným systémom Ubuntu však zostáva základná funkcionalita zachovaná. Je teda možný prístup na vzdialenú plochu, prenos súborov, uzamknutie vzdialeného počítača, posielanie rýchlych správ, zaznamenávanie relácie a funkcia na vymenenie strany s partnerom, kedy sa vymenia role počítačov (ovládajúci počítač sa stane ovládaným). Prenos súborov je zabezpečený dialógovým oknom, kde je prehľadne vidno súborový systém lokálneho aj vzdialeného počítača ako aj front úloh, tak ako je zobrazené v obr. 2.1.




Obr. 2.1: **Grafické rozhranie správcu prenosu súborov v TeamVieweri.** Na ľavej strane je zobrazený súborový systém lokálneho počítača, na pravej strane je súborový systém vzdialeného počítača. V spodnej časti je front úloh, kde sa zobrazí postup prenášaného súboru a v druhej karte *Denník udalostí* je výpis z prenosu súborov.

Chýba ale možnosť pripojiť sa priamo na konzolu vzdialeného počítača, alebo prípadne spúšťať skripty. Táto funkcionalita je pre splnenie navrhnutých podmienok nutne potrebná. Ďalej oproti daným požiadavkám, chýbajú funkcie ako zobrazenie stavu vzdialeného počítača (vyťaženie procesora, pamäti, atp.) a zobrazenie informácií o vzdialenom systéme (hardvérová konfigurácia, verzie nainštalovaného softvéru, atp.).

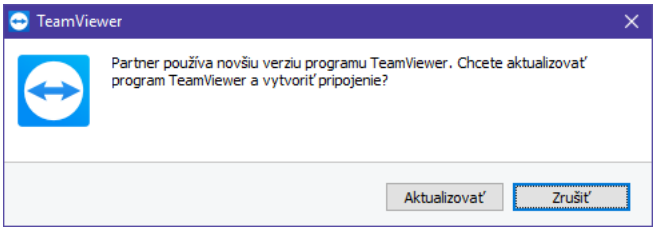
### Možnosť pripojenia sa na vzdialený počítač aj bez nutnosti statickej verejnej IP adresy

Pripojenie k vzdialenému počítaču prebieha zadaním TeamViewer ID počítača, na ktorý sa chceme pripojiť do grafického rozhrania TeamVieweru počítača, z ktorého sa pripájame. TeamViewer ID je 9 až 10 miestny reťazec obsahujúci iba numerické znaky. Je unikátny a jeho hodnota sa pri reštarte počítača nemení [21]. Nie je známy presný algoritmus, pomocou ktorého sa ID generuje, je však isté, že pri generovaní sa používa niekoľko hardverových a softvérových identifikátorov [17]. Následne sa pripojenie verifikuje buď heslom, alebo potvrdením na strane počítača, na ktorý sa pripájame. Heslo môže byť nastavené manuálne, v tomto prípade sa po reštarte počítača nemení, alebo je vygenerované samotným TeamViewerom. V tomto prípade sa heslo zmení pri reštarte počítača, alebo ho môže užívateľ vygenerovať znovu v užívateľskom rozhraní.

### Spoľahlivosť, stabilita a bezpečnosť pripojenia

Spoľahlivosť pripojenia cez TeamViewer na systémoch Ubuntu môže byť diskutabilná. To je ovplyvnené niekoľkými faktormi. Jedným z nich je čiastočná nekompatibilita medzi rôznymi

verziami aplikácie. Je síce možné pripojiť sa pomocou novšej verzie na vzdialený počítač, kde je nainštalovaná staršia verzia, nie je to ale možné opačne. Verzia TeamVieweru na počítači, z ktorého sa pripájame, teda musí byť vyššia, alebo rovnaká, ako verzia nainštalovaná na vzdialenom počítači [18]. Pri pokuse o pripojenie sa na vzdialený systém, kde je verzia TeamVieweru vyššia ako na lokálnom počítači, sa namiesto zobrazenia vzdialenej plochy zobrazí dialógové okno, ktoré vidno na obr. 2.2.




Obr. 2.2: **Dialógové okno oznamujúce konflikt verzií.** Okno bolo vyvolané pri pokuse pripojiť sa na vzdialený počítač, na ktorom bol nainštalovaný TeamViewer verzia 14, pričom na počítači, z ktorého som sa pripájal bola nainštalovaná verzia 13.

Toto môže vytvoriť problém, keďže licencia na komerčné využívanie TeamVieweru je viazaná na verziu softvéru. Pri vydaní novej verzie to teda znamená, že je nutné zakúpiť novú licenciu, alebo sa uistiť, že na vzdialených počítačoch nebude aktualizovaná verzia TeamVieweru.

Ďalší problém som pozoroval v stabilite pripojenia. Pri odhlásení užívateľa na vzdialenom počítači sa okno s pripojenou vzdialenou plochou zatvorí a spojenie sa preruší. Ak sa užívateľ pokúsi v krátkej dobe o pripojenie na rovnaký vzdialený počítač, spojenie sa nenadviaže a následne nie je pripojenie možné, až kým nie je reštartovaný TeamViever démon. Zároveň pri opakovaných pokusoch o pripojenie sa v krátkych intervaloch na vzdialený počítač, ktorého TeamViewer client nemá pripojenie, vzniká zákaz vytvárania akýchkoľvek spojení z počítača, z ktorého nadväzujeme spojenia. Dĺžka tohto zákazu nie je presne známa. Z pozorovania ale usudzujem, že sa jedná o desiatky sekúnd, až minúty.

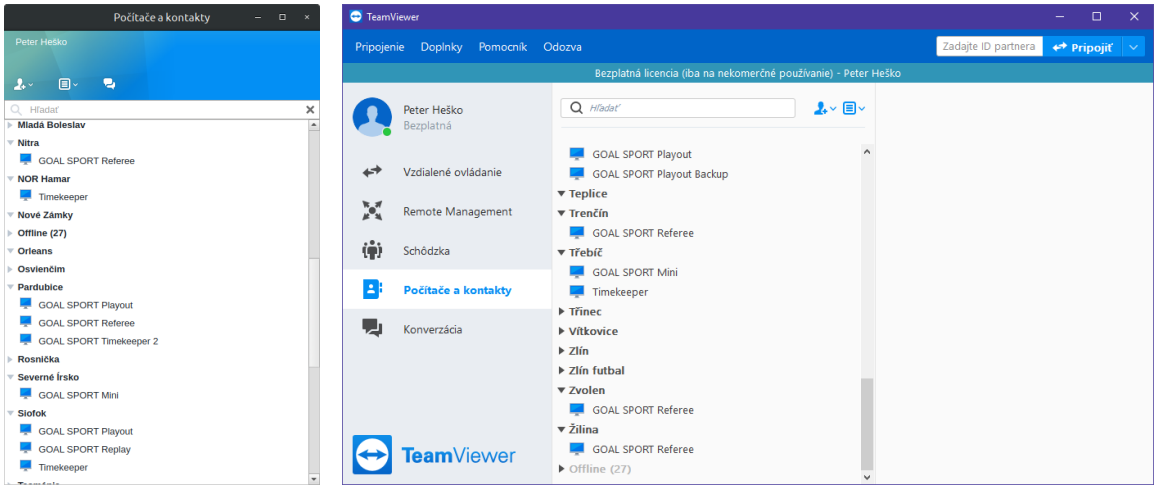
V nastaveniach TeamVieweru je možné určiť, či sa má program spustiť pri štarte systému. V takomto prípade, ak neexistuje konflikt medzi verziami TeamVieweru na jednotlivých počítačoch a TeamViewer démon na vzdialenom počítači beží správne, je možné vytvoriť spojenie na vzdialený počítač kedykoľvek.

Bezpečnosť spojenia je dosiahnutá použitím algoritmu založenom na výmene kľúča pomocou RSA 2048 bit a následne na šifrovaní spojenia pomocou AES 256 bit [20]. Bezpečnosť spojenia je teda podľa dnešných štandardov dostačujúca. Samotný protokol pre prenos dát je proprietárny, je však podobný RDP alebo VNC s prekonávaním NAT a podporou pre prenos súborov, rýchlych správ a ostatných funkcií, ktoré boli spomenuté vyššie. Počítač sa pri spustení TeamViewer démona pripojí na niektorý z tzv. *Master server*, pričom pri pripájaní sa na vzdialený počítač komunikácia prebieha cez tento Master server [6].

### Užívateľské rozhranie

Užívateľské rozhranie pokladám za prijateľne prehľadné a čitateľné. Po vytvorení účtu pre TeamViewer je možné uložiť jednotlivé vzdialené počítače a každému priradiť skupinu, do ktorej patrí. Táto funkcionalita je dostupná aj pre účty s bezplatnou licenciou. Ukladá sa TeamViewer ID vzdialených počítačov, ich heslo, alias a skupina do ktorej patrí. Týmto spô-

sobom sa dá vytvoriť napríklad zoznam lokácií a počítačov, ktoré do nich patria. Následne sa dá na vzdialené počítače pripojiť len niekoľkými kliknutiami. Nie je však možné skupiny do seba navzájom zanárať, alebo počítačom prideliť viac ako jednu skupinu. Vo vytvorenom zozname sa následne zobrazujú iba počítače, na ktorých TeamViewer beží. Všetky počítače, ktoré nie sú pripojené na TeamViewer Master server sa v užívateľskom rozhraní zobrazujú v spoločnej skupine *Offline*. Chýbajú tiež podrobnejšie informácie o vzdialených počítačoch. Bez pripojenia sa na niektorý z nich je možné zistiť len to, či sú, alebo nie sú pripojené na TeamViewer Master server, tzn. či je možné sa na ne pripojiť, alebo nie. Vzhľad a rozloženie užívateľského rozhrania je zobrazený v obr. 2.3.




Obr. 2.3: **Zoznam počítačov v rôznych verziách TeamVieweru.** Na ľavej strane je zoznam vzdialených počítačov vo verzii TeamViewer 13 na operačnom systéme Ubuntu, na pravej strane je vo verzii 14 na operačnom systéme Windows.

Až do verzie TeamViewer 12 sa aplikácia spúšťala s využitím programu Wine, čo mohlo niekedy spôsobovať problémy s kompatibilitou užívateľského rozhrania, alebo dlhou dobou spúšťania programu. Od verzie TeamViewer 13, je ale distribúcia pre Linux natívne podporovaná za pomoci knižnice Qt [19]. To zvýšilo stabilitu a celkovú responzivitu aplikácie.

### Cena

Všetky ceny v tejto časti sú v mene Česká koruna a sú prebraté z oficiálneho cenníka spoločnosti TeamViewer GmbH [14]. TeamViewer je možné používať pod dvomi druhmi licencie. *Firemné/komerčné použitie* a *osobné/nekomerčné použitie*. Osobná licencia je bezplatná, ako však názov naznačuje, je možné TeamViewer pod touto licenciou využívať len na nekomerčné účely. Objavili sa ale aj prípady, kedy softvér chybne detekoval komerčné použitie pri bezplatnej licencii a blokoval, alebo obmedzoval ďalšie pripojenia [23]. Firemná licencia je spoplatnená formou mesačných poplatkov a je rozdelená do 4 cenových kategórií. Najlacnejšia *Business Licence* stojí 639 Kč na mesiac. Je možné naviazať na ňu jeden účet, vytvoriť jedno spojenie súčasne a do zoznamu vzdialených počítačov uložiť do 200 zariadení. *Premium Licence* stojí 1369 Kč na mesiac. Je možné na ňu naviazať viacero účtov, vytvoriť jedno spojenie súčasne a do zoznamu vzdialených počítačov uložiť do 300 zariadení. *Corporate Licence* stojí 2649 Kč mesačne a umožňuje na ňu naviazať viacero účtov, vytvoriť do 3 súčasných spojení a do zoznamu vzdialených zariadení uložiť do 500 počítačov. Najvyššia

licencia *Enterprise Licence* nemá uvedenú cenu, ani konkrétne špecifikácie. Medzi verziami licencií je zachovaná základná funkcionalita popísaná vyššie v tejto práci.

## 2.3 AnyDesk

AnyDesk[3], podobne ako TeamViewer, patrí medzi jedny z najznámejších aplikácií pre prístup na vzdialenú plochu medzi bežnými užívateľmi. Je multiplatformný a je jednoduché ho nainštalovať a používať. Podporuje prepojenie dvoch počítačov s rôznymi operačnými systémami. Je tiež možné sa pripojiť na vzdialený počítač z mobilného zariadenia s operačným systémom Android, alebo iOS. Podporované je tiež spustenie aplikácie bez nutnosti inštalácie na operačnom systéme Windows.

### Kompatibilita s operačným systémom Ubuntu

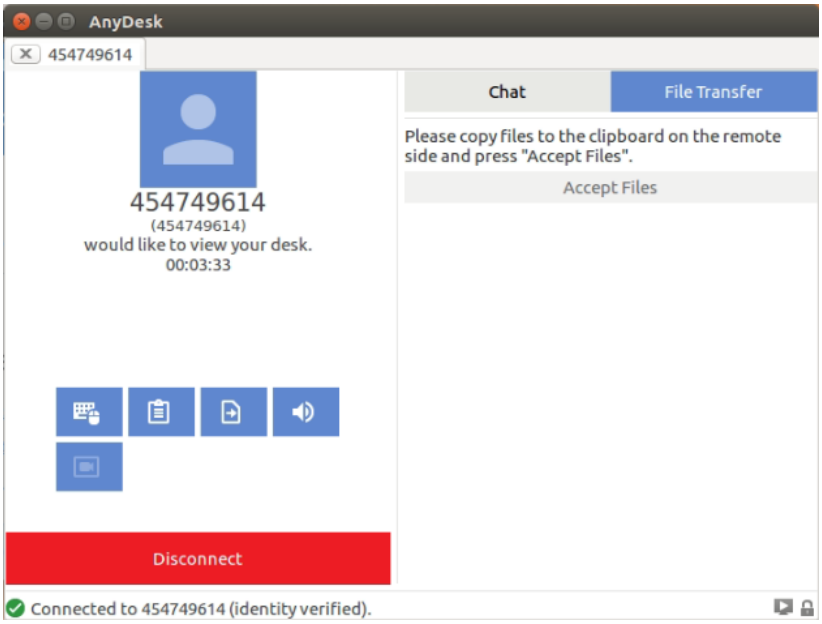
AnyDesk je oficiálne podporovaný pre operačné systémy založené na Linuxe od verzie AnyDesku 2.3.0, ktorá vyšla 9.5.2016 [3]. Súčasná verzia AnyDesku v čase písania tejto práce je 4.0.1. Od začiatku oficiálnej podpory pre systémy založené na Linuxe je podporovaný natívne [9]. Na oficiálnych stránkach je možné stiahnuť balíček deb určený aj pre distribúciu Ubuntu, nie je však špecifikované, ktoré konkrétne verzie sú podporované. Čo sa týka starších verzií, tie na oficiálnych stránkach nie je možné stiahnuť, sú dostupné iba na webových serveroch tretích strán. Osobne som testoval verziu AnyDesku 4.0.0 a 4.0.1 na operačných systémoch Ubuntu 16.04 LTS (Xenial Xerxus) a Ubuntu 18.04 LTS (Bionic Beaver). Obe verzie sú na oboch systémoch plne funkčné.

### Funkcie

Funkcionalita AnyDesku spĺňa hlavné, nie však všetky navrhnuté požiadavky. Je možné pripojiť sa na plochu vzdialeného počítača, ktorú je následne možné ovládať a taktiež je možné prenášať súbory. Na obr. 2.4 je možné vidieť dialógové okno pre prenos súborov. Ďalšia funkcionalita zahŕňa posielanie rýchlych správ na vzdialené počítače, výmena strany so vzdialeným počítačom, na ktorý sme pripojený, zobrazenie informácie o vzdialenom systéme (vybrané informácie o hardvérovej a softvérovej konfigurácii), alebo vytvorenie snímky obrazovky. Je tiež možné vyvolať skratku Ctrl-Alt-Delete, alebo vyvolať reštart vzdialeného počítača. Tieto funkcie však nie sú dostupné, ak na počítači, na ktorý sa pripájame, beží operačný systém Ubuntu.

Popísaný prenos súborov je neprehľadný a môže byť aj nebezpečný, keďže sa môže stať, že v systémovej schránke budú uložené citlivé údaje a to, kedy sa vykoná prenos súborov môže byť potvrdené na oboch stranách spojenia. Ďalej úplne chýba možnosť vytvoriť zoznam vzdialených počítačov. Táto funkcionalita je sprístupnená až na prostrednej verzii platenej licencie (viac v časti 2.3 na strane 11). Ukladajú sa ale informácie o posledných spojeniach. Nie je tiež možné pripojiť sa priamo na konzolu počítača, alebo spúšťať skripty bez pripojenia sa na vzdialenú plochu.

[3] Oficiálna webová stránka aplikácie AnyDesk https://anydesk.com/




Obr. 2.4: **Grafické rozhranie pre prenos súborov pri spojení v AnyDesk.** Prenos v Linuxe funguje tak, že na strane počítača, z ktorého sa pripájame, sa súbor, ktorý chceme preniesť, najprv skopíruje do systémovej schránky a následne sa prenos potvrdí na strane počítača, na ktorý sme pripojený. Potvrdenie sa vykoná kliknutím na tlačidlo *Accept Files*.

### Možnosť pripojenia sa na vzdialený počítač aj bez nutnosti statickej verejnej IP adresy

Informácie v tomto odstavci som prebral z oficiálneho príspevku podpory AnyDesk [2]. Na pripojenie sa k vzdialenému počítaču je potrebné Anydesk ID, alebo AnyDesk Alias a prípadne aj heslo. AnyDesk ID je reťazec, ktorý sa skladá z 9 numerických znakov. Je unikátny a nastavuje sa pri prvom spustení aplikácie. Ak sa vymažú konfiguračné súbory (napríklad pri odinštalovaní aplikácie), identita je permanentne stratená a je nutné vytvoriť nový AnyDesk ID. AnyDesk Alias je taktiež unikátny identifikátor, ktorý je pridelený počítaču pri prvom spustení. Má podobu *name@namespace*, pričom *name* je väčšinou názov počítača a *namespace* má buď hodnotu *ad*, alebo vlastnú – nastavenú užívateľom (dostupné iba pre používateľov, ktorí majú k svojmu účtu priradenú licenciu *Professional*, alebo *Enterprise*). Príklad zobrazenia AnyDesk ID je viditeľný na obr. 2.5. AnyDesk Alias nie je dostupný pre počítače, na ktorých je spustená prenosná verzia softvéru. Užívateľ si môže vybrať, či chce, aby sa identifikátor jeho počítača zobrazoval ako AnyDesk ID, alebo ako AnyDesk Alias. Na pripojenie k vzdialenému počítaču sa teda použije AnyDesk ID, alebo AnyDesk Alias vzdialeného počítača. Ak je na vzdialenom počítači nastavené heslo pre nepotvrdený prístup, je možné sa pomocou tohto hesla autentifikovať. V opačnom prípade je nutné, aby bolo pripojenie potvrdené užívateľom pri vzdialenom počítači. Pre vytvorenie spojenie teda nie je nutná verejná IP adresa.

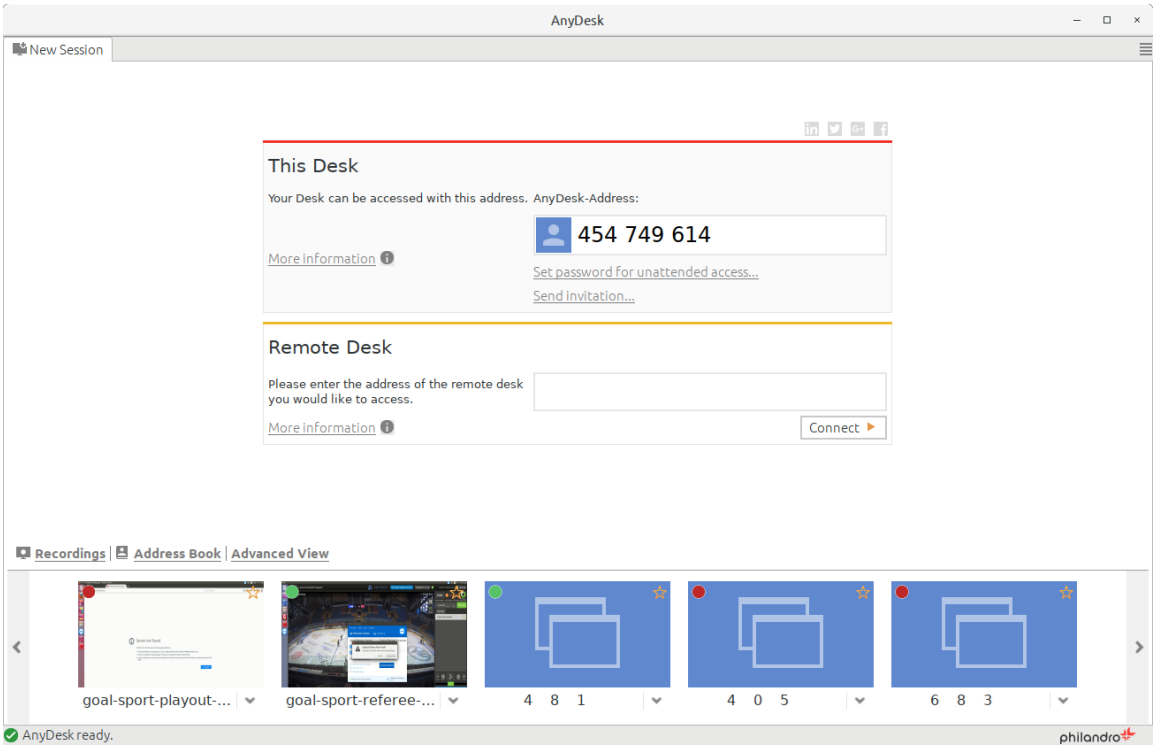
### Spoľahlivosť, stabilita a bezpečnosť pripojenia

AnyDesk je možné nastaviť tak, aby sa zapínal pri spustení počítača. V takom prípade je vždy možné sa na daný počítač pripojiť. Pri používaní aplikácie som nezaznamenal žiadne výpadky spojenia, či nemožnosť pripojiť sa na vzdialený počítač.

Bezpečnosť je dosiahnutá použitím šifrovania založenom na TLS1.2. Prenos dát je zabezpečený pomocou šifrovania AES 128 bit [1].

### Užívateľské rozhranie

Užívateľské rozhranie je jednoduché a prehľadné. Ako som spomínal vyššie, chýba ale možnosť uložiť zoznam vzdialených počítačov, ktorá je odomknutá až pri zakúpení vyššej verzie platenej licencie. Zobrazujú sa ale počítače, na ktoré sme sa nedávno pripojili, ako je možné vidieť na obr 2.5. Pri nich je viditeľný stav pripojenia a korešpondujúci AnyDesk ID alebo AnyDesk Alias. Bez pripojenia sa na vzdialený počítač však nie sú viditeľné žiadne ďalšie informácie.




Obr. 2.5: **Úvodná obrazovka grafického rozhrania AnyDesk.** V hornej časti sa nachádza AnyDesk ID lokálneho počítača. Ten je viditeľný v časti *Toto pracovisko* (na tomto mieste sa prípadne zobrazí AnyDesk Alias). Ak sa chceme pripojiť na vzdialený počítač, zadáme AnyDesk ID, alebo AnyDesk Alias do vstupného poľa v časti *Iné pracovisko*. V spodnej časti je zoznam počítačov, s ktorými bolo nedávno vytvorené spojenie (identifikačné údaje boli z bezpečnostných dôvodov graficky upravené).

### Cena

Údaje uvedené v tejto časti som prebral z oficálneho cenníka spoločnosti AnyDesk Software GmbH [4]. Ceny sú uvedené v mene Americký dolár. AnyDesk je možné používať pod nie-

koľkými licenciami. Licencia s názvom *Free* je bezplatná, avšak s touto licenciou je možné AnyDesk využiť len na občasné osobné použitie. Na komerčné použitie je nutné zaobstarať si platenú licenciu. Existujú 3 druhy platených licencií. Najnižšia verzia s názvom *Lite* stojí 79 USD ročne. Je možné ju priradiť k jednému účtu AnyDesk (každý ďalší priradený účet stojí 29 USD ročne), vytvoriť jedno spojenie so vzdialeným počítačom súčasne, pričom nie je možné dokúpiť ďalšie sloty pre súčasné spojenia. Nie je tiež možné ukladať informácie o vzdialených počítačoch do zoznamu. Stredná verzia licencie s názvom *Professional* stojí 229 USD ročne. K tejto licencii môže byť priradené neobmedzené množstvo účtov AnyDesk, je možné vytvoriť jedno spojenie so vzdialeným počítačom súčasne, pričom každý ďalší slot pre vytvorenie súčasného spojenia stojí 99 USD ročne. Informácie o vzdialených počítačoch je pri použití tejto licentie možné ukladať do zoznamu, pričom nie je obmedzený počet takto uložených vzdialených počítačov. Najvyššia verzia licencie s názvom *Enterprise* nemá uvedenú cenu. Počet účtov AnyDesk, ktoré môžu byť na túto licenciu priradené je neobmedzený. Počet súčasných spojení je označený ako predmet dohody. Aj pri tejto verzii licencie je možné ukladať informácie o vzdialených počítačoch do zoznamu, pričom počet takto uložených zariadení nie je obmedzený.

## 2.4 Voľne dostupné riešenia

Na správu a monitorovanie vzdialených počítačov existuje niekoľko voľne dostupných nástrojov. Väčšina takýchto softvérových riešení ale spĺňa len časť požadovanej funkcionality. Preto je v tomto prípade potrebné použiť niekoľko programov s rôznymi funkciami, aby bola požadovaná funkcionalita dosiahnutá. V tejto sekcii sa zameriam na nástroje, ktoré patria medzi najpoužívanejšie na danej platforme. Medzi ne patria x11vnc, Remmina, Nautilus, openssh, autossh a rôzne nástroje na zobrazenie softvérovej a hardvérovej konfigurácie, ako aj nástroje na zobrazenie aktuálneho vyťaženia jednotlivých komponentov.

### Kompatibilita s operačným systémom Ubuntu

Keďže sú predmetom skúmania nástroje na správu a monitorovanie počítačov s operačným systémom Ubuntu 16.04, nebudem sa zaoberať nástrojmi, ktoré nie sú kompatibilné s daným operačným systémom. Všetky nástroje popísané v tejto sekcii sú teda kompatibilné s Ubuntu 16.04.

### Funkcie

Ako je spomínané vyššie, funkcionalita je rozdelená do niekoľkých nástrojov. Zabezpečený prístup na terminál vzdialeného počítača je možné dosiahnuť pomocou programu *OpenSSH*[4]. Na vzdialenom počítači v tomto prípade musí bežať *OpenSSH server*, ktorý je spustený ako služba, čo zabezpečí možnosť pripojenia sa aj po reštartovaní vzdialeného počítača. Na systéme, z ktorého sa pripájame sa použije program *OpenSSH client*.

Zabezpečený prístup na plochu vzdialeného systému je možný pomocou ľubovoľnej aplikácie fungujúcej ako VNC klient. Príkladom môže byť *TigerVNC*, *VNC Viewer*, alebo *Remmina*[5], na ktorú sa v tejto práci zamerám. Remmina dokáže okrem iného vytvoriť spojenie protokolu VNC, čo umožní zobraziť a ovládať vzdialenú plochu. Protokol VNC ale nie je sám o sebe zabezpečený. Na dosiahnutie zabezpečeného zobrazenia a ovládania

[4]Oficiálna webová stránka programu OpenSSH https://www.openssh.com/
[5]Oficiálna webová stránka aplikácie Remmina https://remmina.org/

vzdialenej plochy je nutné použiť SSH tunel. V tomto prípade môžeme v Remmine nastaviť, aby automaticky vytvorila a následne sa najprv pripojila na daný SSH tunel a potom cez tento tunel nadviazala VNC spojenie. Na splnenie tejto funkcionality musí byť na strane vzdialeného počítača spustený SSH server, kvôli vytvoreniu SSH tunela, a program *x11vnc*[6]. Ten plní úlohu VNC servera[13] a je možné nakonfigurovať systémovú službu, ktorá bude zabezpečovať, aby bol program spustený neustále, aj po reštartovaní vzdialeného počítača.

Funkciu zabezpečeného prenosu súborov dosiahneme použitím protokolu SFTP. Na strane vzdialeného počítača je požadované len spustenie programu OpenSSH server, ktorý zahŕňa aj SFTP server. Na strane počítača, z ktorého sa pripájame, potrebujeme spustiť ľubovoľného SFTP klienta. Môžeme teda napríklad použiť konzolový program *sftp*, alebo sa na SFTP server pripojiť pomocou súborového prehliadača *Nautilus*.

Informácie o softvérovej a hardvérovej konfigurácii, ako aj aktuálne údaje o zaťažení jednotlivých komponentov vzdialeného počítača je možné získať pomocou systémových nástrojov, ktoré sú väčšinou súčasťou štandardnej inštalácie Ubuntu. Po pripojení sa na vzdialený terminál pomocou SSH môžeme tieto nástroje na vzdialenom systéme spúšťať, pričom sa na štandardný výstup vypíše ich výsledok. Samozrejme môžeme údaje čítať aj priamo z kernelových dátových štruktúr, ktoré sú zvyčajne pripojené automaticky operačným systémom do pseudo súborového systému */proc*. Dostupné nástroje ale často ponúkajú prehľadnejšie podanie danej informácie. Medzi tieto nástroje napríklad patrí *free*, ktorý vypíše informácie o pamäti v systéme, *uptime*, ktorý vypíše dobu behu systému a *nproc*, ktorý vypíše počet dostupných centrálnych procesorových jednotiek. Na získanie informácií o využití procesoru môžeme použiť obsah súboru *stat*, ktorý sa nachádza v pseudo súborovom systéme */proc*. Podrobné údaje o grafických kartách značky Nvidia ponúka nástroj *nvidia-smi*, ktorý je automaticky nainštalovaný pri inštalácii ovládača pre grafické karty značky Nvidia.

Keďže sa jedná o niekoľko aplikácií a nie jednu, ktorá by zahŕňala všetky funkcie, nie je možné uložiť a spravovať jednotný zoznam vzdialených počítačov, ktorý by sa dal následne použiť na rýchle využívanie jednotlivých funkcií.

### Možnosť pripojenia sa na vzdialený počítač aj bez nutnosti statickej verejnej IP adresy

Jednotlivé nástroje sami o sebe nedisponujú vlastnosťami, ktoré by im umožňovali prekonanie NAT a teda pripojenia sa na vzdialený počítač, ktorý nemá verejnú statickú IP adresu. Toto vyplýva z faktu, že všetká funkcionalita závisí od SSH spojenia za pomoci programov openssh server a openssh client. Takéto spojenie nedokáže prekonať NAT, avšak existujú technológie, ktoré tento problém odstránia. Dané technológie a ich použitie je popísané v kapitole 3 na strane 16.

### Spoľahlivosť, stabilita a bezpečnosť pripojenia

Aby sme zabezpečili možnosť pripojenia sa na vzdialený počítač vždy, keď je zapnutý a pripojený do siete, musíme sa uistiť, že potrebné programy plniace úlohu servera na vzdialenom počítači sú spustené neustále a po zapnutí vzdialeného počítača sa automaticky spustia. Pri SSH serveri nie je nutná dodatočná konfigurácia, keďže predvolené nastavenie openssh server spúšťa SSH server ako systémovú službu. V tomto prípade teda operačný systém zabezpečí neustály chod serveru, aj po reštartovaní vzdialeného počítača. Ohľadom VNC serveru existuje viacero možností. Program, ktorý plní úlohu VNC servera, môžeme

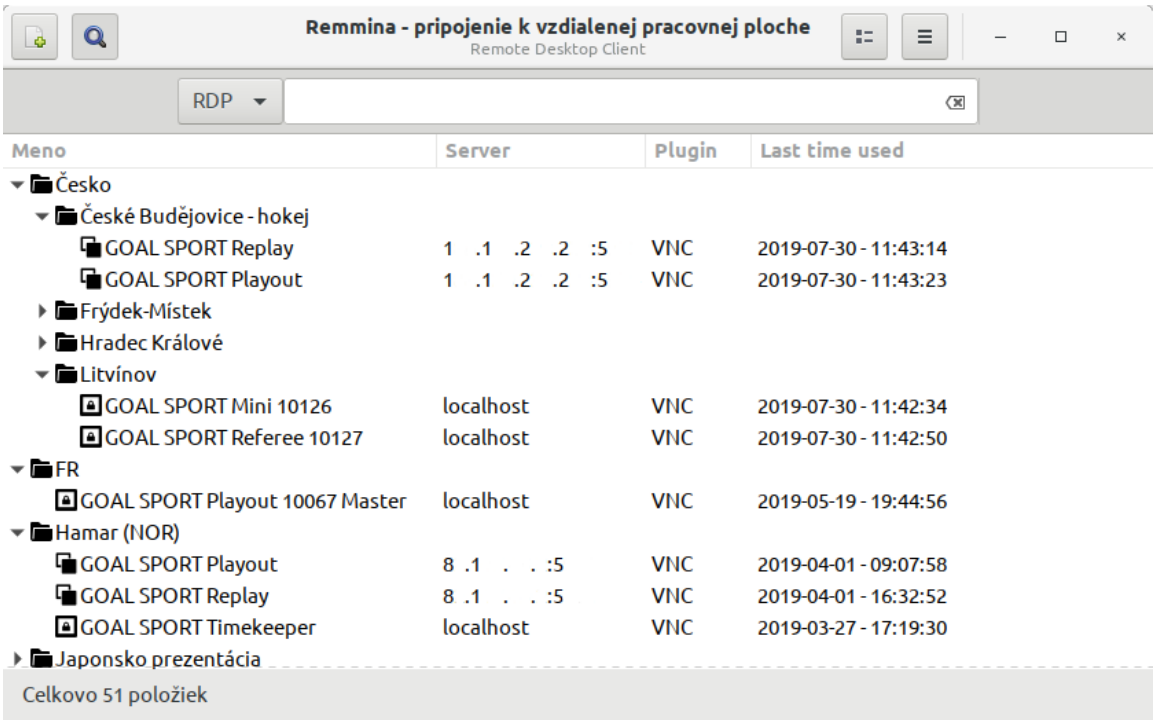
[6] Oficiálna webová stránka programu x11vnc http://www.karlrunge.com/x11vnc/

spúšťať pri štarte vzdialeného počítača pomocou démona *cron*. Server zostane spustený, ak je pri spustení x11vnc použitý argument forever. Ďalšia možnosť je vytvorenie systémovej služby, ktorá zabezpečí chod VNC serveru, podobne ako je to v prípade systémovej služby zabezpečujúcej chod SSH serveru.

SSH spojenie za pomoci openssh je zabezpečené, čo vyplýva z faktu, že protokol SSH je definovaný ako zabezpečený protokol. Na zabezpečenie prenosu dát cez VNC protokol môžeme použiť tunelovanie cez SSL/TSL, je však odporúčané využiť tunelovanie cez SSH [13]. Pri použití VNC protokolu cez SSH tunel je teda aj toto pripojenie zabezpečené. Pri pozorovaní stability spojenia som nezaznamenal žiadne nedokonalosti.

### Užívateľské rozhranie

Na spustenie openssh client môžeme použiť ľubovoľný terminál, napríklad gnome-terminal, ktorý je v Ubuntu predvolený. Tento terminál disponuje okrem základnej funkcionality tiež možnosťou upraviť jeho vzhľad, čo je vítaná vlastnosť. Pri nadväzovaní SFTP spojenia môžeme využiť terminálový program *sftp*, alebo napríklad predvolený prehliadač súborov v Ubuntu *Nautilus*. *Nautilus* má prehľadné rozhranie, na SFTP server sa môžeme pripojiť priamo v prehliadači, alebo spustením *Nautilus* z terminálu, kde v argumente uvedieme cestu k SFTP serveru. *Remmina*, v tomto prípade ako sprostredkovateľ VNC spojenia, má prehľadné užívateľské rozhranie. Údaje o pripojení k vzdialeným počítačom môžeme ukladať do zoznamu, ktorý môže byť rozdelený na skupiny, pričom jednotlivé skupiny môžu obsahovať ďalšie podskupiny. Zobrazenie zoznamu, respektíve stromu obsahujúceho jednotlivé záznamy je možné vidieť na obr. 2.6. Správa zoznamu je jednoduchá, pri pridávaní nového záznamu, alebo jeho editácii sa používa dialógové okno s parametrami daného záznamu.




Obr. 2.6: **Grafické užívateľské rozhranie aplikácie Remina.** V hlavnom okne je viditeľný zoznam uložených konfigurácií pre pripojenie k jednotlivým vzdialeným počítačom (niektoré údaje boli z bezpečnostných dôvodov graficky upravené).

Z dôvodu decentralizácie jednotlivých funkcií do rôznych programov môže nastať neprehľadnosť. Tento fakt tiež spôsobuje neefektivitu práce pri správe a monitorovaní vzdialených systémov, keďže na súčasné splnenie všetkých bodov funkcionality je nutné spustenie niekoľkých aplikácií. Údaje o pripojení k vzdialeným počítačom sa dajú uložiť do zoznamu iba v niektorých nástrojoch. Správa takéhoto zoznamu je teda komplikovaná a neprehľadná.

### Cena

Remmina, x11vnc aj openssh sú slobodné softvéry. Remmina má licenciu GLP[10], x11vnc má taktiež licenciu GLP[5] a openssh má licenciu BSD[8]. Na používanie týchto nástrojov teda nie sú nutné žiadne náklady.

# Kapitola 3

# Technológie pre pripojenie k uzlom bez verejnej IP adresy

Predpokladajme, že počítač, z ktorého sa chceme pripájať nazveme lokálny počítač. Systém, na ktorý sa chceme pripojiť nazveme vzdialený počítač. Ak lokálny, alebo vzdialený, alebo oba počítače nemajú verejnú statickú IP adresu, nastáva problém, kedy nástroje na správu a monitorovanie vzdialených systémov popísané v sekcii 2.4 na strane 12, nebudú fungovať. Táto kapitola sa teda zaoberá technológiami, ktoré by mohli byť použité na prekonanie NAT a tým problém odstrániť. Existuje viacero metód, ako dosiahnuť prekonanie NAT, zameriam sa však na dve z nich, ktoré sú pre túto prácu najrelevantnejšie.

## 3.1 Reverzný SSH tunel

Informácie k tejto časti som čerpal z blogu popisujúcom vytvorenie reverzného SSH tunela [12]. Ak potrebujeme prístup na vzdialený počítač, ktorý nemá verejnú IP adresu a pripájame sa z počítača, ktorý verejnú IP adresu má, jedno z riešení je reverzný SSH tunel. Ako názov napovedá, jedná sa o SSH tunel, ktorý je vytvorený zo strany vzdialeného počítača. Znamená to teda, že ak máme k dispozícii počítač s verejnou IP adresou, môžeme sa naň pripojiť pomocou reverzného SSH tunelu z počítača, ktorý verejnú IP adresu nemá. Počítač s verejnou IP adresou budem v tejto práci nazývať relay server. Následne bude vzdialený počítač s neverejnou IP adresou prístupný cez otvorený reverzný tunel. Ak sa ale potrebujeme pripájať z počítača, ktorý taktiež nemá verejnú adresu, môžme nechať na relay serveri otvorený niektorý z portov a následne sa pripájať na daný port z lokálneho počítača s neverejnou adresou. Relay server následne spojenie prepojí na otvorený reverzný tunel, cez ktorý sa dostaneme na vzdialený počítač.

### Vytvorenie reverzného tunela

Pre vytvorenie reverzného tunela na vzdialenom počítači môžme použiť program *openssh*. Príkaz, ktorým sa reverzný tunel spustí, je rozdelený na niekoľko častí. Prvou je samotný program, ktorý spustíme, v tomto prípade je to `ssh`. Nasledujú argumenty `-fN`, kde `f` spôsobí, že sa program spustí na pozadí a argument `N`, že sa nebudú spúšťať žiadne príkazy na počítači, na ktorý sa pripájame. Argument `N` je v tomto prípade vhodný aj z toho dôvodu, že nechceme vytvoriť klasické SSH spojenie, kde by sme posielali príkazy, ale len smerovať porty, čím sa vytvorí reverzný tunel. Nasleduje argument `-R 10022:localhost:22`. Argument `R` zabezpečí vytvorenie reverzného tunela. Hodnota argumentu `10022:localhost:22`

určuje vlastnosti reverzného tunela. V tomto prípade dané hodnoty znamenajú, že relay server bude smerovať všetkú komunikáciu na svojom porte `10022` na port `22` vzdialeného počítača. Na záver sú ako argument určené prihlasovacie údaje na SSH server bežiaci na relay serveri. V tomto príklade je to `relay_user@1.1.1.1`, kde predpokladáme, že `relay_user` je meno užívateľa na relay serveri a `1.1.1.1` je verejná statická IP adresa relay servera. Vo výsledku by príkaz na vytvorenie reverzného tunela mohol vyzerať nasledovne:

```
ssh -fN -R 10022:localhost:22 relayserver_user@1.1.1.1
```

Pri spustení tohto príkazu by bolo možné pripojiť sa z relay serveru pomocou SSH na vzdialený počítač, nie však priamo z lokálneho počítača na vzdialený počítač. Pripojenie z relay serveru by mohlo byť dosiahnuté nasledovným príkazom:

```
ssh vzdialeny_user@localhost -p 10022
```

kde `vzdialeny_user` je meno užívateľa na vzdialenom počítači a adresa `localhost` znamená, že sa pripojí na svoju vlastnú adresu na porte `10022` – pripojenie na reverzný tunel. Samozrejme by bolo možné najprv manuálne nadviazať spojenie z lokálneho počítača na relay server a potom sa pripojiť na vytvorený tunel. To je však zdĺhavé, keďže je nutné SSH spojenie vytvoriť dva krát.

Riešením je upraviť konfiguráciu SSH serveru na relay serveri a následne upraviť argument príkazu na vytvorenie reverzného tunela spúšťaného na vzdialenom počítači. Súbor obsahujúci konfiguráciu SSH serveru je predvolene uložený v `/etc/ssh/sshd_conf`. V tomto súbore je nutné pridať voľbu `GatewayPorts clientspecified`. Táto voľba zabezpečí aby SSH server na relay serveri povolil pripojenie na porty, presmerované vzdialeným počítačom pomocou reverzného tunalu, aj iným vzdialeným počítačom. Pre túto voľbu je predvolená hodnota `no`, čo sposobí, že na presmerované porty je možné sa pripojiť iba z relay serveru. Príkaz na vytvorenie reverzného tunela sa potom upraví do nasledujúcej podoby:

```
ssh -fN -R 1.1.1.1:10022:localhost:22 relay_user@1.1.1.1
```

Zmena teda nastala iba v hodnote argumentu `-R`. Oproti predchádzajúcemu príkladu sa teraz koniec reverzného tunela nachádza na adrese `1.1.1.1:10022`. V tomto stave je teda možné sa na vzdialený počítač pripojiť priamo z lokálneho počítača pomocou príkazu:

```
ssh vzdialeny_user@1.1.1.1 -p 10022
```

Ak by sme preložili význam tohto príkazu, môžeme povedať, že sa teda pripájame z lokálneho počítača na koniec reverzného tunelu, otvoreného medzi vzdialeným počítačom a relay serverom.

Na zachovanie tunela pri výpadku pripojenia k internetu môžeme použiť program *autossh*. Ten sa pokúsi po výpadku spojenia toto spojenie znovu nadviazať. Aby sme zachovali spojenie aj po reštarovaní počítača, môžme využiť démona *cron* tak, že príkaz so všetkými potrebnými argumentmi vložíme do súboru `/etc/crontab`.

## 3.2 VPN

Informácie k tejto sekcii som čerpal z návodu na vytvorenie, konfiguráciu a správu VPN pomocou programu *OpenVPN* [11]. Podobne ako použitie reverzného tunelu, aj použitie protokolu VPN dokáže odstrániť problém, ktorý vzniká pri absencii statickej verejnej IP adresy. Metóda, ktorou ale tento problém rieši je úplne odlišná. Funguje na princípe vytvorenia virtuálnej privátnej siete, čo teda odstráni potrebu vzdialených počítačov na statickú verejnú IP adresu. Týmto spôsobom sa teda efektívne prekoná NAT.

Jeden z najpoužívanejších programov implementujúcich VPN je *OpenVPN*. Tento sofvér je voľne dostupný, vrátane zdrojového kódu a aj práve preto som ho vybral ako príklad na vytvorenie VPN založenej na architektúre klient/server. Táto technológia vyžaduje vytvo-

renie OpenVPN serveru, ktorý musí byť dostupný z internetu, čo znamená, že je potrebné, aby mal statickú verejnú IP adresu. Jednotliví klienti sa následne pripájajú na tento server pomocou daného softvéru. Pri pripájaní prebehne obojstranné overenie certifikátov. To znamená, že server overí certifikát pripájaného klienta a klient overí certifikát serveru, na ktorý sa pripája. Následne sa klient pripojí na server, čím v požadovanom prípade sprístupní seba ostatným klientom pripojeným na server a zároveň je mu umožnený prístup k daným klientom. Pravidlá, ktorí klienti sa môžu a nemôžu pripojovať na jednotlivých ostatných klientov, sú možné konfigurovať. Všetká následná komunikácia medzi jednotlivými stanicami je kryptograficky zabezpečená.

### Vytvorenie a správa VPN

Existujú dva spôsoby, ako vytvoriť virtuálnu lokalnu sieť. Prvý je ethernetový most. Pre dosiahnutie funkčnosti tejto metódy je potrebné vytvoriť virtuálne rozhranie, ktoré bude následne pomocou mostu prepojené s fyzickým rozhraním a teda lokálnou sieťou. Tomuto rozhraniu je pridelený rozsah voľných IP adries z lokálnej siete, ktoré sú pridelené ostatným klientom pripojeným na VPN server. Použitie ethernetového mostu z pohľadu klienta teda priradí jednu IP adresu z lokálnej siete každému klientovi pripojenému na VPN server. Vytvorí sa tak dojem, že pripojení klienti sú v podstate pripojení priamo do lokálnej siete počítača. Výhodou je, že jednotlivé lokálne siete pridané do virtuálnej privátnej siete nemusia byť prekonfigurované tak, aby každá z nich mala unikátnu IP adresu.

Druhý spôsob je použitie smerovania. V tomto prípade je nutné zaistiť, aby boli adresy lokálnych sietí pripájaných klientov unikátne. Pri nenaplnení tejto požiadavky by mohol nastať konflikt IP adries. To vyplýva z faktu, že pri použití tejto metódy sú jednotlivé lokálne siete spojené virtuálnym smerovačom. Nevýhodou teda je, že pri nasadzovaní VPN pomocou tohto spôsobu na už existujúce siete je nutná rekonfigurácia daných sietí, pretože väčšinou nie je relevantný dôvod, aby IP adresa lokálnej siete bola zmenená z predvoleného nastavania. Zároveň vytvorenie a konfigurácia takejto virtuálnej privátnej siete je oveľa zložitejšia, ako pri použití ethernetového mostu. Na druhej strane ponúka väčšiu kontrolu nad pravidlami, ktoré určujú možnosti prepojenia jednotlivých klientov. Táto metóda je tiež o niečo pomalšia, keďže sa pri prepojení dvoch lokálnych sietí používa L3 smerovanie.

Pri oboch metódach je možné, aby VPN bola spustená na smerovači, alebo priamo na vzdialených počítačoch. Ak by sme ale chceli na pripojenie sa do virtuálnej privátnej siete použiť smerovač, musíme si najprv overiť, či daný smerovač túto funkcionalitu podporuje. Nie všetky smerovače totiž podporujú pripojenie do VPN.

# Kapitola 4

# Návrh softvéru podľa požadaviek firmy Daite s.r.o.

Táto kapitola sa zaoberá návrhom aplikácie pre zabezpečený vzdialený prístup a správu počítačov podľa popísaných požiadaviek spoločnosti Daite s.r.o. Dané požiadavky sú špecifikované v sekcii 2.1 na strane 3. V prvej sekcii sa zaoberám výberom relevantnej technológie, ktorá bude použitá na elimináciu problému spôsobeného neprítomnosťou statickej verejnej IP adresy na koncových zariadeniach. Ďalej sa venujem výberu a použitiu nástrojov, ktoré budú schopné splniť funkcionalitu špecifikovanú v požiadavkách. Na záver popisujem dôvod výberu prostredia, ktoré bude použité na implementáciu výsledného nástroja.

## 4.1 Použitie technológie pre prekonanie NAT

V kapitole 3 na strane 16 som popísal dve najrelevantnejšie technológie na prekonanie NAT, vzhľadom k zadaným požiadávkam. Aj keď je možností viacero, spomínané dva spôsoby sú najzmysluplnejšie a oba by sa dali pri implementácii použiť. Pri ich analýze som ale usúdil, že použitie reverzného SSH tunela má viacero výhod oproti použitiu virtuálnej privátnej siete.

### Spoločné vlastnosti

Dané nástroje majú niektoré atribúty podobné, čo vyplýva z faktu, že oba spĺňajú zadané požiadavky. Patrí medzi ne napríklad počet možných súčasne pripojených vzdialených počítačov na server so statickou verejnou IP adresou, ktorý sa teoreticky síce líši, avšak v praxi je podobný. Strop vytvára hardverové obmedzenie serveru. Oba nástroje na splnenie požiadavky prekonávať NAT potrebujú aspoň jeden systém, ktorý disponuje statickou verejnou IP adresou. To vyplýva z charakteru problému, keďže pripojenie na vzdialený počítač je možné iba cez internet a lokálne adresy teda nemôžu byť relevantný cieľ pripojenia. Prenos dát je pri oboch riešeniach šifrovaný, čo umožňuje zabezpečenú komunikáciu.

### Výhody reverzného SSH tunela oproti použitiu VPN

Hlavnou výhodou použitia reverzného SSH tunela je fakt, že na jeho vytvorenie je potrebný len program OpenSSH. Je nutné, aby pre splnenie požiadavky popisujúcej pripojenie sa na vzdialený terminál, bol na vzdialenom systéme spustený SSH server. V prípade použitia VPN by bola potrebná inštalácia a konfigurácia ďalšieho nástroja, čo pri nasadzovaní

vzdialených systémov násobí náročnosť a čas potrebný na správne nastavenie všetkej funkcionality. V tomto prípade teda môže OpenSSH plniť funkciu pripojenia sa na vzdialený terminál, ako aj funkciu možnosti pripojenia sa na vzdialený systém bez nutnosti statickej verejnej IP adresy na strane vzdialeného a zároveň aj lokálneho počítača.

Oproti konfigurácii OpenVPN je konfigurácia reverzného SSH tunelu jednoduchá. Pri OpenVPN je nutné najprv správne zvoliť metódu pripájania vzdialených lokálnych sietí, ktorá sa použije a následne správne nakonfigurovať jednotlivých klientov a server. To môže zahŕňať konfiguráciu virtuálnych sieťových zariadení, nastavovanie smerovacích pravidiel, alebo relatívne zložitú správu certifikátov a určovanie pravidiel pre jednotlivých užívateľov. Na druhej strane konfigurácia a správa reverzného SSH tunela zahŕňa nastavenie SSH serveru na relay serveri, pričom na dosiahnutie funkčnosti je nutné v jeho konfigurácii zmeniť len jednu hodnotu a na strane vzdialených systémov je len potrebné zaistiť, aby bol reverzný tunel vždy otvorený. Na to je možné využiť už existujúce nástroje ako autossh a cron. Správa užívateľov a ich práv môže byť následne riešená pomocou systému správy užívateľov v operačnom systéme.

Nasadenie novej inštalácie vzdialených systémov z pohľadu systémovej konfigurácie je taktiež zložitejšie pri VPN. V závislosti na metóde nasadenia VPN môže byť potrebné navrhnúť podrobnú štruktúru lokálnej siete. Toto môže byť ďalej sťažené faktom, že vzdialené systémy sú často nasadzované do lokálnych sietí, ktorých správcom je tretia strana. Pri nasadzovaní VPN na už existujúce inštalácie môže vzniknúť požiadavka na prerobenie štruktúry lokálnej siete, alebo v prípade, že by sme chceli použiť smerovač ako OpenVPN klienta, mohla by nastať nekompatibilita zariadení. Nasadenie reverzného SSH tunela je na rozdiel od VPN jednoduché. Nie je nutné riešiť štruktúru lokálnej siete, ani kompatibilitu zariadení. Nasadenie prebieha rovnako pre nové inštalácie, ako aj pre tie už existujúce.

Pri pripájaní sa z lokálneho na vzdialený počítač je pri reverznom tuneli nutný iba nástroj OpenSSH klient. Nie je v podstate nutná žiadna konfigurácia lokálneho počítača, aby sme sa mohli pripojiť na vzdialený počítač. V tomto prípade OpenVPN vyžaduje konfiguráciu lokálneho počítača rovnakým spôsobom, ako sa konfigurujú vzdialené počítače.

## 4.2 Nástroje na splnenie funkcionality

V kapitole 2 na strane 12 som analyzoval dostupné riešenia, pričom som sa zameral na splnenie podmienok funkcionality jednotlivými nástrojmi. Dospel som k výsledku, že neexistuje riešenie pozostávajúce z jedného nástroja, ktorý by úplne spĺňoval všetky požiadavky na funkcionalitu. Navyše, aj pri snahe nájsť čo najjednoduchšie riešenie formou použitia multifunkčného nástroja ako TeamViewer, alebo AnyDesk som narazil na problém. Ten spočíva v tom, že ich funkcionalita je obmedzená a pre potreby firmy Daite s.r.o. by bolo nutné zakúpenie licencie. Aj keď by bolo možné funkcionalitu rozšíriť, napríklad pomocou zabudovania daného nástroja do komplexnejšieho systému, návrh a následná správa takéhoto systému by mohla byť veľmi obtiažna, keďže sa jedná o softvér, ktorý nie je voľne šíriteľný.

Z týchto dôvodov som sa rozhodol využiť voľne dostupné nástroje. Ich funkcionalita je síce rozdelená medzi niekoľkými softvérovými riešeniami, ale v prípade tejto práce to považujem za výhodu, keďže to v podstate znamená, že môžem určiť presnú funkcionalitu výsledného systému, ktorý bude tieto riešenia zahrňovať. V prípade potreby bude tiež možné funkcionalitu pridať, alebo odobrať, vzniká tak väčšia flexibilita pri plnení požadovanej funkcionality.

Na prístup na vzdialený terminál teda použijem OpenSSH, ako najrozšírenejšiu implementáciu protokolu SSH. Na pripájanie sa na vzdialenú plochu a jej ovládanie využijem

protokol VNC. Dôvodom je fakt, že je to protokol priamo navrhnutý na zobrazenie a ovládanie vzdialenej plochy cez internet. Ako VNC server použijem x11vnc, keďže sa jedná o jednu z najrozšírenejších implementácií protokolu VNC pre x11 server. Pripojenie na VNC server a zabezpečenie tohto pripojenia dosiahnem použitím programu Remmina. Tá je zahrnutá v štandardnej inštalácii súčasne podporovaných verzií Ubuntu. Remmina tiež dokáže spojenie zabezpečiť pomocou vytvorenia reverzného SSH tunelu, cez ktorý následne naviaže VNC spojenie. Pri tomto riešení je teda potrebný len otvorený reverzný SSH tunel zo vzdialeného počítača a spustený VNC server na vzdialenom počítači. K splneniu požiadavky na zabezpečený prenos súborov som sa rozhodol využiť protokol SFTP. SFTP server je súčasťou OpenSSH, nie je teda potrebná dodatočná inštalácia a konfigurácia ďalších nástrojov na vzdialenom počítači. Na lokálnom systéme využijem program na prehliadanie súborového systému Nautilus, ktorý v tomto prípade dokáže fungovať aj ako SFTP klient. Nautilus, podobne ako Remmina, je zahrnutý v štandardnej inštalácii aktuálne podporovaných verzií Ubuntu a je tiež predvoleným súborovým prehliadačom.

Na zbieranie dát o softvérovej a hardvérovej konfigurácii vzdialených systémov, ako aj údaje o zaťažení ich jednotlivých komponentov použijem systémové nástroje na toto určené. Spúšťanie jednotlivých nástrojov bude prebiehať cez SSH spojenie, takže ich výstup bude cez toto spojenie dostupný. Používať budem systémové nástroje analyzované a popísané v sekcii 2.4 na strane 12.

## 4.3 Implementačné prostredie

Ako implementačné prostredie som sa rozhodol využiť aplikačný rámec *Qt*[1]. K tomuto rozhodnutiu som dospel na základe nasledujúcich faktov:

**Predošlé skúsenosti** S aplikačným rámcom Qt som sa stretol v priebehu predmetu Tvorba užívateľských rozhraní. Mám teda základnú predstavu, ako daný aplikačný rámec funguje a aká je jeho štruktúra. Po skúsenosti s prácou v Qt som usúdil, že by mohol byť ideálny na riešenie tejto práce.

**Ideálny na tvorbu užívateľského rozhrania** Qt je navrhnutý ako multiplatformný aplikačný rámec so zameraním na jednoduché a intuitívne vytváranie užívateľského rozhrania [7]. Má vlastné integrované vývojové prostredie *Qt Creator*, ktoré okrem iného obsahuje aj grafický editor vzhľadu vyvíjanej aplikácie, kde je možné upravovať jednotlivé grafické komponenty a meniť ich vlastnosti. To umožňuje v krátkom čase vytvoriť funkčný prototyp aplikácie. Oproti iným aplikačným rámcom, ktoré podobnú funckiu neponúkajú je to veľká výhoda.

**Zahrnuté knižnice** Okrem vlastného integrovaného vývojového prostredia ponúka Qt aj ďalšiu výhodu. Tou je veľké množstvo zahrnutých knižníc, plniacich široké spektrum funkcií. Dané knižnice sú pravidelne aktualizované a často implementujú najnovšie algoritmy.

**Veľké množstvo užívateľov** Qt je používaný veľkým množstvom užívateľov, či už sa jedná o jednotlivcov, alebo celé firmy. Z tejto popularity vyplýva fakt, že na internete sa nachádza nespočetne veľa článkov, návodov a diskusných fór, ktorých hlavnou témou je programovanie aplikácií s využitím aplikačného rámca Qt. Taktiež je dobre spracovaná

[1] Oficiálna webová stránka aplikačného rámca Qt https://www.qt.io/

dokumentácia k aplikačnému rámcu a jeho knižniciam, čo len prispieva k celkovej jednoduchosti použitia.

# Kapitola 5

# Implementácia softvéru

Navrhnutý softvér som implementoval vytvorením aplikácie *Support tool*. Tá sa skladá z dvoch častí. Prvá z častí je klientská, druhá serverová. V tejto kapitole sa venujem popisu používania a implementácie oboch častí.

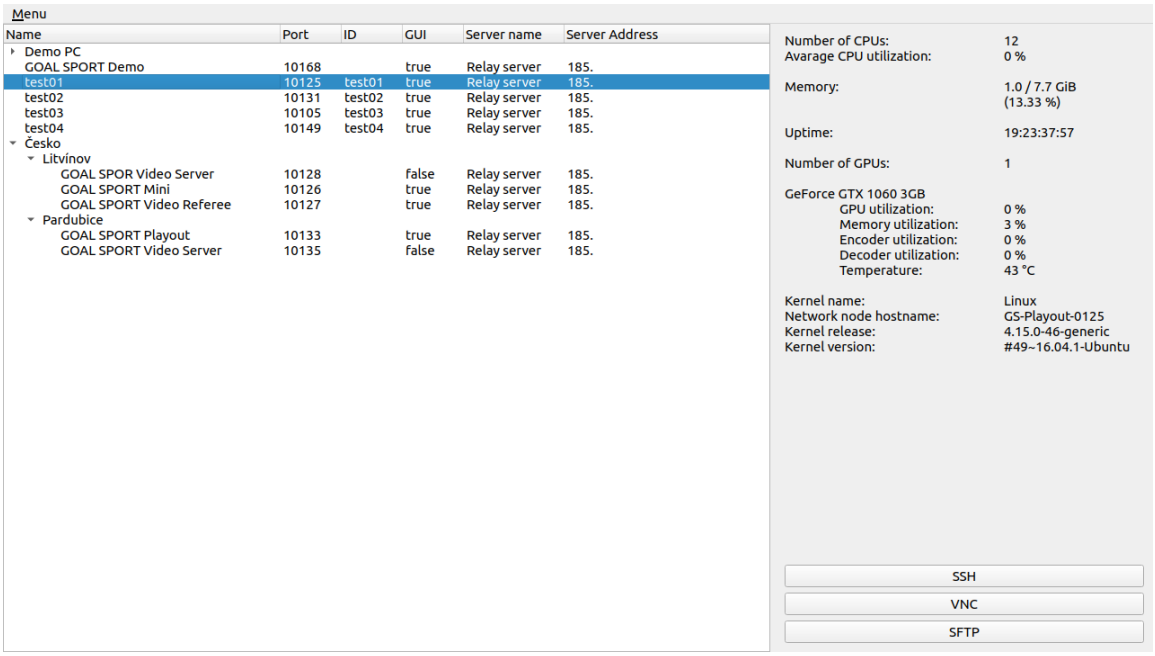
## 5.1 Klient

### Grafické rozhranie

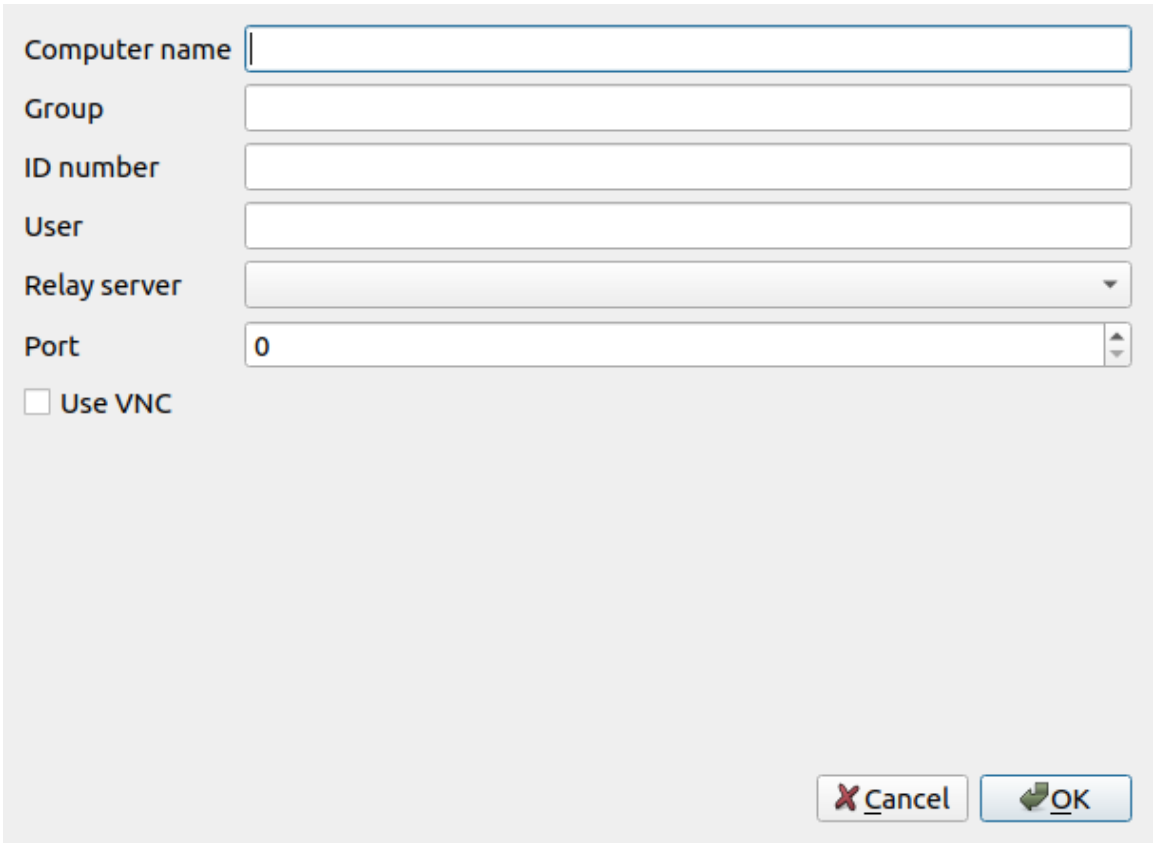
Implementácia grafického rozhrania bola relatívne jednoduchá vďaka nástrojom, ktoré sú v integrovanom vývojovom prostredí Qt Creator poskytnuté. Ako je možné vidieť na obr. 5.1, grafické rozhranie je rozdelené na dve hlavné časti: na ľavej strane sa nachádza stromová štruktúra obsahujúca skupiny, podskupiny a záznamy o pripojení k vzdialeným počítačom, pravá strana sa zaoberá konkrétnym počítačom, ktorý je v stromovej štruktúre aktuálne vybratý. Na tejto strane sa v hornej časti nachádzajú informácie o softvérovej a hardvérovej konfigurácii, ako aj aktuálne údaje o zaťažení jednotlivých komponentov, alebo doba prevádzky vzdialeného počítača. V spodnej časti pravej strany sa nachádzajú tri tlačidlá, ktoré spúšťajú externé nástroje s korešpondujúcimi argumentami.

Stromová štruktúra zobrazuje niekoľko údajov o jednotlivých skupinách, podskupinách a záznamoch o pripojení k vzdialeným počítačom. Prvý stĺpec zobrazuje uložený názov danej položky. Tento údaj má len informačný význam, nepoužíva sa pri monitorovaní, alebo spúšťaní externých nástrojov. Druhý stĺpec obsahuje port, na ktorom je pripojený reverzný SSH tunel k relay serveru. Tretí stĺpec obsahuje ID vzdialeného počítača. Podobne ako prvý stĺpec, aj táto položka má len informačný charakter. Štvrtý stĺpec obsahuje boolovskú hodnotu, ktorá určuje, či daný vzdialený počítač má užívateľské rozhranie, alebo nie. Piaty stĺpec obsahuje názov relay serveru, ktorý je uložený v nastaveniach aplikácie. Táto hodnota má informačný charakter, slúži na lepšie rozlíšenie jednotlivých relay serverov. Posledný, šiesty stĺpec obsahuje IP adresu relay serveru. Táto hodnota sa používa na monitorovanie daného vzdialeného počítača, ako aj na spúšťanie externých nástrojov s korešpondujúcimi argumentami.

V ľavom hornom rohu sa nachádza položka *Menu*. Tá obsahuje tri akcie: pridanie nového vzdialeného počítača, otvorenie nastavení Support tool klienta a vypnutie aplikácie. Pridanie nového počítača otvorí dialógové okno zobrazené na obr. 5.2. Jednotlivé riadky v tomto dialógovom okne významovo korešpondujú so stĺpcami v stromovej štruktúre.

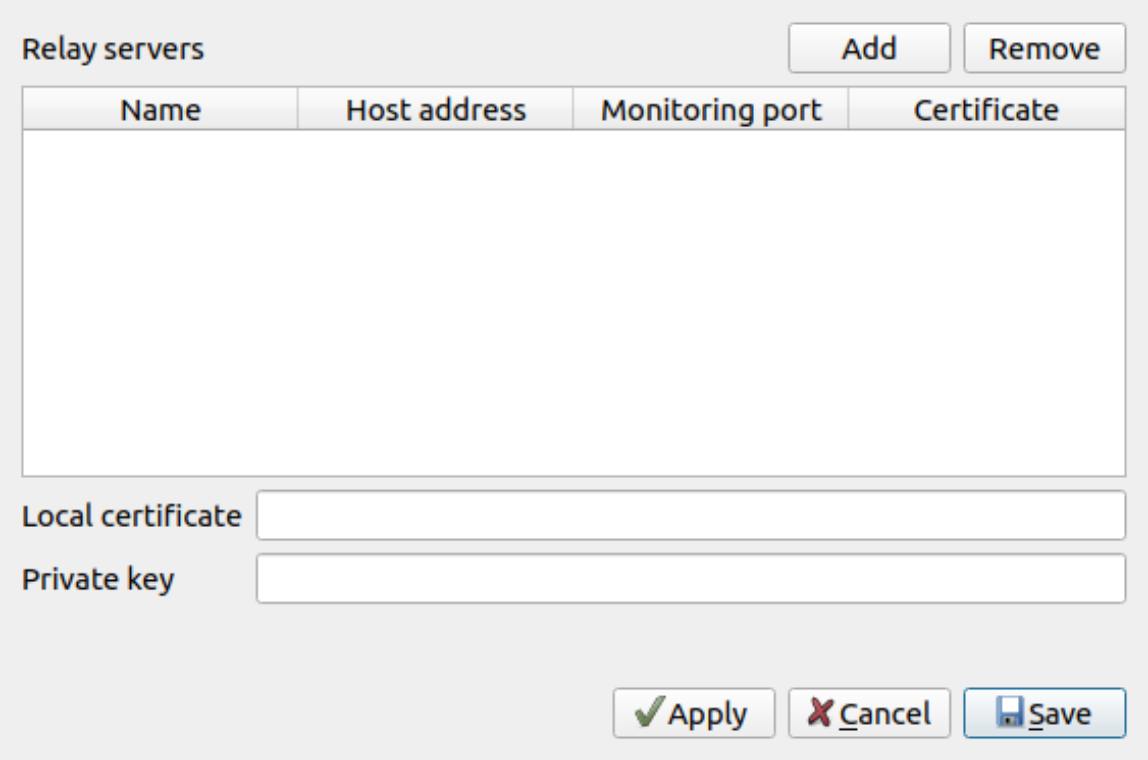



Obr. 5.1: **Grafické užívateľské rozhranie implementovaného nástroja Support tool.** (niektoré údaje boli z bezpečnostných dôvodov graficky upravené).




Obr. 5.2: **Dialógové okno na pridanie nového vzdialeného počítača.**

Nastavenia aplikácie sú tiež upravované pomocou dialógového okna. Toto okno je zobrazené na obr. 5.3, kde môžeme vidieť tabuľku, ktorá obsahuje údaje o relay serveroch, tlačidlo na pridanie nového relay serveru, tlačidlo na odobratie vybratého relay serveru a dva riadky. V prvom je uložená cesta k lokálnemu certifikátu, v druhom cesta k privátnemu kľúču. Tabuľka relay serverov obsahuje štyri stĺpce. Prvý stĺpec obsahuje názov serveru. Táto hodnota má informačný charakter na lepšie rozlíšenie jednotlivých serverov. Druhý stĺpec obsahuje adresu relay serveru. Tá sa používa v kombinácii s portom konkrétneho vzdialeného počítača na pripojenie sa naň. Tretí stĺpec obsahuje číslo portu, na ktorom počúva Support tool server spustený na relay serveri. Posledný stĺpec obsahuje cestu k súboru, v ktorom je uložený SSL certifikát serveru.




Obr. 5.3: **Dialógové okno na pridanie nového vzdialeného počítača.**

### Implementácia

Support tool klient je implementovaný ako Qt Widget aplikácia. Všetky grafické prvky sú teda vytvorené pomocou knižníc Qt. Ihneď po spustení programu sa načítajú konfiguračné súbory. Ich cesta je pevná, nastavená na `$HOME/.config/support-tool`. Konfiguračné súbory sú dva. Jeden pre ukladanie záznamov do stromovej štruktúry, druhý na ukladanie užívateľských nastavení. Oba súbory sú vo formáte JSON.

Zobrazenie stromovej štruktúry je dosiahnuté rekurzívnym načítaním daného konfiguračného súboru, následným vytvorením stromového modelu a jeho naplnenie načítanými dátami a nakoniec nastavením tohto modelu na stromové zobrazenie.

Asi najzaujímavejšia časť tohto programu je pripojenie sa na relay server, načítanie dát o vzdialenom počítači a ich zobrazenie. Všetko sa deje pri kliknutí do stromovej štruktúry na niektorý zo záznamov o vzdialenom počítači. Najprv sa ukončí predchádzajúce spojenie so

serverom, ak existuje, následne sa skontrolujú informácie o údajoch, ktoré sú k pripojeniu nevyhnutné. Ak sú všetky údaje v poriadku, vytvorí sa nový objekt triedy `QSslSocket`. Táto trieda je z knižnice s rovnakým názvom a implementuje SSL soket. Tomuto soketu sa následne nastavia požadované certifikáty a zároveň sa soket nastaví do režimu, kedy bude vyžadovať overenie serveru. Následne sa pokúsi o pripojenie na Support tool server. Proces čaká, kým bude naviazané šifrované spojenie. Ak sa spojenie vytvorilo bez chyby, klient pošle serveru správu obsahujúcu číslo portu vzdialeného počítača. Server potom pošle jednu z troch možných odpovedí. `NM` znamená, že vzdialený počítač nie je monitorovaný Support tool serverom. `OF` znamená, že vzdialený počítač je síce v zozname monitorovaných počítačov na Support tool serveri, server naň ale nemá žiadne pripojenie. V prípade tretej možnosti server postupne po riadkoch v dohodnutom poradí odošle údaje o softvérovej a hardvérovej konfigurácii a aktuálne údaje o vyťažení systemu, klient ich následne v rovnakom poradí spracuje a zobrazí v užívateľskom rozhraní

## 5.2 Server

Support tool server je terminálová aplikácia spúšťaná na relay serveri, ktorá monitoruje a po dobu behu uchováva údaje o vzdialených počítačoch pripojených na relay server.

### Implementácia

Ihneď po štarte sa spustí načítavanie súboru obsahujúceho zoznam vzdialených počítačov a adresy koncov reverzných SSH tunelov, cez ktoré sú tieto vzdialené počítače na relay server pripojené. Ak je na začiatku riadku tohto súboru mriežka, riadok je ignorovaný. Po načítaní údajov o vzdialených počítačoch, ktoré sa budú monitorovať sa pre každý počítač vytvorí objekt triedy `RemotePC`, ktorý bude uchovávať informácie, ktoré z nich Support tool server získa. Ak sa načítal aspoň jeden údaj o vzdialenom počítači, vytvoria sa objekty triedy `WorkerThread`, ktoré plnia funkciu opakovaného paralelného spúšťania skriptu na získanie informácií o vzdialenom počítači. Pre každý monitorovaný vzdialený počítač sa vytvorí práve jeden objekt triedy `WorkerThread`. Tento objekt následne spustí metódu na získanie informácii o vzdialenom počítači. Táto metóda najprv z dát uložených v objektoch triedy `RemotePC` zistí, či bol pri poslednom spustení metódy vzdialený počítač online. Ak bol offline, metóda skúsi, či už je online. Ak je stále offline, metóda bude čakať 30 sekúnd, kým sa znovu pozrie na stav vzdialeného počítača. Ak odpovie na výzvu, znamená to, že už nie je offline a spustí sa teda prvý skript na získanie informácií, ktoré sa počas behu vzdialeného systému nebudú meniť. Medzi tieto informácie patrí počet centrálnych výpočetných jednotiek, veľkosť inštalovanej pamäti RAM, softvérová konfigurácia operačného systému, počet grafických kariet značky Nvidia a ich produktové názvy. Ak sa pri spustení metódy zistí, že vzdialený počítač bol online aj pri poslednom spustení danej metódy, spustí sa druhý skript, ktorý získa informácie o aktuálnom zaťažení systémových komponentov. Medzi tieto informácie patrí zaťaženie procesora, množstvo voľnej pamäte RAM, počet inštalovaných grafických kariet značky Nvidia, vyťaženie jednotlivých grafických kariet, ich pamäte, enkodéru a dekodéru a ich teplota a celkový čas prevádzky systému.

Oba skripty fungujú s využitím tzv. *Here document*, ktorý uloží obsah skriptu do premennej a následne je spustený proces, ktorý vytvorí SSH spojenie, kde sa ako argument použije názov tejto premennej. Výsledkom je teda spustenie Bash skriptu na vzdialenom počítači pomocou SSH spojenia. Druhý skript je ale ešte zaujímavý tým, že údaje o vyťažení procesoru sa dajú získať len vo forme počtu milisekúnd od štartu počítača v jednotlivých

režimoch procesoru. To znamená, že na získanie priemerného vyťaženia procesoru musíme tento údaj získať 2 krát. V druhom skripte sa teda hneď na začiatku vypíše údaj o dobe procesoru v jednotlivých stavoch, uloží sa čas, kedy bol tento údaj prečítaný a pokračuje sa so získavaním ostatných údajov. Keďže počet grafických kariet je premenná a získaní informácií je pomerne zdĺhavé, vypočíta sa čas, ako dlho musí skript čakať, aby pri druhom čítaní údajov o procesore prešlo aspoň približne 5 sekúnd. Ak načítavanie ostatných údajov trvalo dlhšie ako 5 sekúnd, údaje o procesore sa prečítajú ihneď. Vypísané údaje sú potom spracované a uložené do príslušných objektov triedy `RemotePC` a objekt triedy `WorkerThread` emituje signál, že boli načítané nové dáta.

Pri novom pripojení klienta na server si server prečíta požiadavku v podobe čísla portu vzdialeného počítača. Na túto žiadosť následne odpovie adekvátnym stavom vzdialeného počítača a zároveň spojí číslo sledovaného portu so soketom, cez ktorý je klient pripojený pomocou objektu `QMap`. Na signál, ktorý je emitovaný pri načítaní nových dát je pripojený slot, ktorý zistí, či sú práve načítané dáta sledované nejakým klientom a ak áno, odošle ich na soket, cez ktorý je klient pripojený.

# Kapitola 6

# Testovanie implementovaného softvéru

Aplikácia Support tool spĺňa požadovanú funkcionalitu. Vyplýva to z faktu, že boli použité nástroje, ktoré dokopy funkcionalitu spĺňali. Keďže Support tool tieto nástroje spája, je zaručené dosiahnutie požadovanej funkcionality.

### Zabezpečenie spojenia

Spojenie medzi vzdialenými počítačmi a relay serverom je vytvárané pomocou reverzného SSH tunela, čo znamená, že je použité SSH spojenie. To je samo o sebe považované za zabezpečené spojenie. Spojenie medzi Support tool klientom a Support tool serverom je riešené pomocou SSL serveru a SSL soketu. SSL server sa spúšťa na Support tool serveri a je implementovaný ako potomok triedy `QTcpServer`. Na oboch stranách komunikácie sa následne vytvoria objekty triedy `QSslSocket`, ktorým sa nastavia potrebné certifikáty na overenie opačnej strany. Po tomto bode z pohľadu programovania prebieha rovnaká komunikácia ako pri použití triedy `QTcpSocket`, avšak pri pozorovaní obsahu komunikácie cez program *Wireshark* som zistil, že prenášané dáta sú šifrované a teda nečitateľné bez prístupu k privátnym kľúčom.

### Použitie Support tool klienta

Pri prvom spustení Support tool klienta je potrebné vykonať niekoľko krokov, aby bolo možné sa pripojiť na server. Budeme uvažovať, že Support tool server je už spustený na relay serveri a je pripravený odpovedať na dotazy klientov.

Prvým krokom je otvoriť nastavenia klienta a pridať nový relay server. Do stĺpca *Host address* vyplníme doménové meno servera, alebo jeho verejnú statickú IP adresu. Následne vyplníme stĺpec *Port*, ktorý zodpovedá číslu portu, na ktorom Support tool server počúva prichádzajúce spojenia. Nakoniec je nutné zadať cestu k súboru, ktorý obsahuje SSL certifikát serveru. Túto informáciu zadáme do stĺpca *Certificate*. Stĺpec *Name* je ľubovoľný. Následne je pre úspešné overenie klienta serverom nutné nastaviť cesty k súborom obsahujúcim lokálny certifikát klienta a jeho privátny kľúč. Dané údaje zadáme do im odpovedajúcich polí *Local certificate* a *Private key*. Po uložení nastavení môžeme pridať nový vzdialený počítač. Pomocou dialógového okna vyplníme požadované údaje. Ak v poli *Group* napíšeme medzi názvy skupín lomítko, skupina, ktorá je na pravej strane od lomítka sa stane podskupinou a skupina na ľavej strane od lomítka. Ak vieme, že vzdialený počítač má na-

inštalované grafické rozhranie, môžeme zaškrtnúť box *Use VNC*. Potvrdením dialógového okna sa daný záznam vytvorí a priradí do stromovej štruktúry na miesto špecifikované poľom Group. Tento počítač teraz môžeme editovať dvojitým kliknutím naň. To otvorí dialógové okno podobné dialógovému oknu na pridávanie počítačov, je tu však navyše tlačidlo *Delete*, ktoré editovaný záznam o vzdialenom počítači vymaže zo stromovej štruktúry. Pri potvrdení tohto dialógového okna sa v zázname aktualizujú údaje podľa toho, aké údaje boli v dialógovom okne editácie záznamu pri jeho potvrdení. Pri kliknutí na záznam sa na pravej strane vypíše buď, že počítač nie je monitorovaný (čo znamená, že Support tool server o ňom nezbiera informácie), počítač je offline (čo znamená, že Support tool server sa o ňom snaží zbierať dáta, avšak nie je možné sa naň pripojiť), alebo sa vypíšu údaje o softvérovej a hardvérovej konfigurácii a zároveň aktuálny stav využitia jednotlivých komponentov. Pri kliknutí na záznam v stromovej štruktúre sa tiež aktivujú tri tlačidlá v pravom dolnom rohu podľa toho, aké údaje záznam obsahuje. Tlačidlo *SSH* spustí gnome-terminal s SSH klientom pripájajúcim sa na daný vzdialený počítač. Tlačidlo *VNC* spustí program Remmina s pripojením na daný vzdialený počítač. Tlačidlo *SFTP* spustí program Nautilus a otvorí v ňom SFTP spojenie na daný vzdialený počítač.

### Použitie Support tool serveru

Server neukladá žiadne konfiguračné súbory na disk. Cesty k všetkým potrebným súborom sú mu predané pri spustení pomocou argumentov. Program Support tool server má štyri argumenty, pričom všetky sú vyžadované. Sú to nasledovné argumenty:

- r – –remote-ports Cesta k súboru, ktorý obsahuje údaje o vzdialených počítačoch, ktoré budú monitorované. Na každom riadku je jeden záznam v tvare

  `address:remote_port remote_user`,

  kde `address:remote_port` je adresa a port reverzného SSH tunelu a `remote_user` je meno užívateľa na vzdialenom počítači. Riadok je ignorovaný, ak je na jeho začiatku mriežka.

- c – –certificates Cesta k priečinku, ktorý obsahuje súbory s lokálnym SSL certifikátom a privátnym kľúčom. Súbor s lokálnym SSL certifikátom musí mať názov `server_local.pem` a súbor s privátnym kľučom `server_local.key`.

- p – –monitoring-port Číslo portu, na ktorom bude Support tool server počúvať prichádzajúce spojenia od Support tool klienta.

- u – –user-certs Cesta k priečinku, ktorý obsahuje verejné SSL certifikáty užívateľov, ktorí majú mať prístup k Support tool serveru.

Po špecifikovaní jednotlivých súborov môžeme Support tool server spustiť. Ten začne počúvať na špecifikovanom porte a odpovedať dotazom klientov.

# Kapitola 7

# Záver

Cieľom mojej bakalárskej práce bolo analyzovať dostupné riešenia pre správu a monitorovanie vzdialených systémov, preskúmať technológie použiteľné pri implementácii vlastného riešenia a dané riešenie implementovať. V druhej kapitole som zistil, že komerčné riešenia daného problému sú často pre účel tejto bakalárskej práce drahé a aj napriek tomu nespĺňajú požadovanú funkcionalitu. Analyzované voľne dostupné riešenia na druhej strane nie sú finančne náročne, avšak ich funkcionalita je decentralizovaná na niekoľko nástrojov. Z toho vyplýva, že ich používanie je menej efektívne, pretože je nutné namiesto jedného nástroja, použiť niekoľko nástrojov. V ďalšej kapitole som sa zaoberal skúmaniu technológií na prepojenie dvoch počítačov, z ktorých jeden, alebo oba nemajú statickú verejnú IP adresu. V tejto kapitole sú teda podrobne preskúmané dve technológie, ktoré by v tejto práci mohli byť použité pri implementácii vlastného nástroja. V štvrtej kapitole som vybral jednu z dvoch spomínaných technológii na prekonanie NAT. Zistil som, že pre riešenie tejto práce má SSH spojenie pomocou reverzného SSH tunela oveľa viac výhod ako VPN. Následne som vybral jednotlivé implementácie voľne dostupných riešení a implementačné prostredie pre vlastný nástroj. Zvolil som mnohé z riešení spomínaných v druhej kapitole, pretože boli od začiatku vybrané ako vhodní kandidáti na náhradu komerčných riešení. Ako implementačné prostredie som zvolil aplikačný rámec Qt, vďaka jeho unikátnym vlastnostiam pri tvorbe užívateľského rozhrania a širokej škále dostupných knižníc. Navrhnutý nástroj som následne implementoval do podoby aplikácie s názvom Support tool. Implementácia je popísaná v kapitole číslo 5. Vytvorený nástroj som následne otestoval a zistil som, že spĺňa zadané požiadavky. Samozrejme sa jedná len o prototyp aplikácie, dokázal som však, že spojením voľne dostupných riešení do komplexnejšieho systému môže byť nahradená, dokonca obohatená funkcionalita komerčných riešení. Ako ďalšie vylepšenia vytvoreného softvéru by mohla byť optimalizácia algoritmu, ktorým sa získavajú dáta, alebo dlhodobé uchovávanie dát o vzdialených systémoch. Podobné nástroje majú takmer vždy čo zlepšovať, keďže platí, že pri väčšom množstve relevantných a prehľadných informácií sa zvyšuje rýchlosť riešenia závad na vzdialených, ale aj lokálnych systémoch.

# Literatúra


[1] Alonso, M.: *How to Improve the Security Settings in your AnyDesk Account.* September 2016, [Online; navštívené 17.1.2019].
URL https://blog.anydesk.com/blog-detail/how-to-improve-security-settings-in-your-anydesk-account

[2] AnyDesk Software GmbH: *AnyDesk ID and Alias.* [Online; navštívené 17.1.2019].
URL https://support.anydesk.com/AnyDesk_ID_and_Alias

[3] AnyDesk Software GmbH: *Full changelog.* [Online; navštívené 16.1.2019].
URL https://download.anydesk.com/changelog.txt

[4] AnyDesk Software GmbH: *Pricing that suits your needs.* [Online; navštívené 18.1.2019].
URL https://anydesk.com/order

[5] BIZX, LLC: *LibVNCServer.* [Online; navštívené 24.4.2019].
URL https://sourceforge.net/projects/libvncserver/

[6] Braden, T.: *TeamViewer authentication protocol.* Január 2013, [Online; navštívené 10.1.2019].
URL https://www.optiv.com/blog/teamviewer-authentication-protocol-part-1-of-3

[7] Company, T. Q.: *This is Qt.* [Online; navštívené 25.5.2019].
URL https://www.qt.io/what-is-qt/

[8] Developers, O.: *Copyright Policy.* [Online; navštívené 22.1.2019].
URL https://www.openbsd.org/policy.html

[9] Ebersol, A.: *AnyDesk: Better Than TeamViewer?* Júl 2017, [Online; navštívené 16.1.2019].
URL https://pclosmag.com/html/Issues/201707/page03.html

[10] Gatta, A.: *A feature rich Remote Desktop Application.* [Online; navštívené 22.1.2019].
URL https://remmina.org/

[11] INC., O.: *2x HOW TO.* [Online; navštívené 3.4.2019].
URL https://openvpn.net/community-resources/how-to/

[12] Nanni, D.: *How to access a Linux server behind NAT via reverse SSH tunnel.* Máj 2015, [Online; navštívené 27.1.2019].
URL http://xmodulo.com/access-linux-server-behind-nat-reverse-ssh-tunnel.html

[13] Runge, K.: *x11vnc: a VNC server for real X displays.* [Online; navštívené 22.1.2019].
URL http://www.karlrunge.com/x11vnc/

[14] TeamViewer GmbH: *Get Your TeamViewer Subscription Today!* [Online; navštívené 12.1.2019].
URL https://www.teamviewer.com/en/buy-now/

[15] TeamViewer GmbH: *Previous versions.* [Online; navštívené 8.1.2019].
URL https://www.teamviewer.com/en/download/previous-versions/

[16] TeamViewer GmbH: *TeamViewer 9.x, 8.x, 7.x, 6.x, 5.x.* [Online; navštívené 8.1.2019].
URL https://www.teamviewer.com/en/download/old-versions.aspx

[17] TeamViewer GmbH: *Existing Teamviewer ID on new computer?* Apríl 2017, [Online; navštívené 8.1.2019].
URL https://community.teamviewer.com/t5/TeamViewer-General/Existing-Teamviewer-ID-on-new-computer/td-p/5543

[18] TeamViewer GmbH: *Is there a way to support someone that has latest version with my older version?* Apríl 2017, [Online; navštívené 10.1.2019].
URL https://community.teamviewer.com/t5/TeamViewer-General/Is-there-a-way-to-support-someone-that-has-latest-version-with/td-p/6631

[19] TeamViewer GmbH: *[Update] TeamViewer 13.* December 2017, [Online; navštívené 12.1.2019].
URL https://community.teamviewer.com/t5/Linux/Update-TeamViewer-13/td-p/24537

[20] TeamViewer GmbH: *How secure is TeamViewer?* Apríl 2018, [Online; navštívené 10.1.2019].
URL https://community.teamviewer.com/t5/Knowledge-Base/How-secure-is-TeamViewer/ta-p/4619

[21] TeamViewer GmbH: *What is a TeamViewer ID?* Január 2019, [Online; navštívené 8.1.2019].
URL https://community.teamviewer.com/t5/Knowledge-Base/What-is-a-TeamViewer-ID/ta-p/49515

[22] TeamViewer GmbH: *Which operating systems are supported.* Január 2019, [Online; navštívené 8.1.2019].
URL https://community.teamviewer.com/t5/Knowledge-Base/Which-operating-systems-are-supported/ta-p/24141

[23] wrswaim: *TeamViewer thinks I'm using the software commercially!* [Online; navštívené 28.4.2019].
URL https://community.teamviewer.com/t5/TeamViewer-General/TeamViewer-thinks-I-m-using-the-software-commercially/td-p/8284/page/8