Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra informatiky a kvantitativních metod

Distribuovaný informační a diagnostický systém sítě

Diplomová práce

Autor: Martin Kašpar
Studijní obor: Informační management (IM5)

Vedoucí práce: Ing., Pavel Kříž, Ph.D.

Hradec Králové Listopad 2014

Prohlášení:

Prohlašuji, že jsem tuto diplomovou práci pod vedením Ing. Pavla Kříže, Ph.D. vypracoval samostatně a v seznamu použité literatury jsem uvedl všechny odborné zdroje a literaturu, z kterých jsem čerpal.

vlastnoruční podpis

V Hradci Králové dne 14.11.2014 Martin Kašpar

Poděkování:

Děkuji vedoucímu diplomové práce Ing. Pavlu Křížovi, Ph.D. za metodické vedení práce a trpělivost při našich konzultacích.

Anotace

Diplomová práce pojednává o obecném fungování počítačových sítí a principech jejich diagnostiky. Popsána je zde hlavně struktura počítačové sítě, síťové prvky, ze kterých se skládá, a směrování zajišťující volbu trasy přenášených dat. Na základě analýzy současného stavu sítě HKfree ve své praktické části představuje návrh na zlepšení situace v oblasti monitoringu, diagnostiky a informovanosti v síti. Tento návrh je následně zpracován formou klientské aplikace a její serverové podpůrné části. Řešení je nejdříve popsáno, následují ukázky implementace a nakonec je testováno. Význam práce spočívá v řešení problému reálného a fungujícího spolku HKFree, kterému výsledek již nyní přináší užitek. Tento užitek má tendenci se stupňovat na základě hlubšího testování a zpětné vazby.

Klíčová slova

Počítačová síť, komunitní bezdrátová síť, aplikace, peer-to-peer, monitoring, diagnostika, informovanost

Annotation

Distributed informational and diagnostic network system

This Master Thesis deals with the general functionality of the computer networks and with the principles of their diagnostics. The main focus is on the structure of computer networks, elements of these networks, which is the network composed of and also on the routing securing the route choice of the transferred data. Based on the analysis of the current state of the HKfree network this thesis introduces the proposal on how to improve monitoring, diagnostics and awareness in the network. Subsequently these suggestions are processed into the client application and it supportive server part. Application is in the first described in detail, then follow illustrations of implementation and in the end comes the testing of the application itself. Main contribution of this thesis is based in solving the real-life issue of the community network HKfree. In trial run these findings have already been implemented and proved themselves useful. It is strongly expected that the gains of this project will tend to enhance as the project is put into the operation publicly and the feedback of users is collected.

Key words

Computer network, community wireless network, application, peer-to-peer, monitoring, diagnostics, awareness

Obsah

1 Seznam použitých zkratek

AS - Autonomní systém (skupina směrovačů a IP prefixů se společnou směrovací politikou a pod společnou správou)

BGP - Border Gateway Protocol (dynamický směrovací protokol)

FTP - File Transfer Protocol (Protokol pro přenos souborů mezi počítači)

HTTP - Hypertext Transfer Protocol (Protokol pro výměnu hypertextových dokumentů ve formátu HTML)

ICMP - Internet Control Message Protocol (Protokol pro odesílání chybových zpráv)

IP - IP adresa (jednoznačná identifikace zařízení)

ISO - International Organization for Standardization (Světová federace národních normalizačních organizací)

ISP - Internet Access Provider (Poskytovatel internetového připojení)

LAN - Local Area Network (Lokální počítačová síť)

MAN - Metropolitan Area Network (Městská počítačová síť)

NAT - Network Address Translation (Překlad síťových adres)

OSI - Open Systems Interconnection (Propojení otevřených systémů)

OSPF - Open Shortes Path First protocol (dynamický směrovací protokol)

RIP - Routing Information Protocol (dynamický směrovací protokol)

SO - Správce oblasti

TCP - Transmission Control Protocol (protokol transportní vrstvy)

TTL - Time To Live (parametr životnosti paketů)

UDP - User Datagram Protocol (protokol transportní vrstvy)

UPS - Uninterruptible Power supply Source (Nepřerušitelný zdroj energie)

WAN - Wide Area Network (Rozlehlá počítačová síť)

WWW - World Wide Web (soustava propojených hypertextových dokumentů)

2 Úvod

Počítačové sítě v dnešní době představují nedílnou součást každodenního života. Zprostředkovávají přenos informací nejen v různých oblastech vědy, hospodářství, průmyslu, obchodu ale i zábavy a mezilidské komunikace. V této práci bude přiblížena samotná podstata fungování počítačových sítí pro snadnější pochopení, jak jsou data přenášena a zpracovávána. Pokud je tento proces z nějakého důvodu přerušen, je nutné diagnostikovat a odstranit příčinu pro obnovení funkčnosti. Budou proto představeny i metody diagnostiky nefunkčnosti počítačové sítě.

V praktické části se práce bude zabývat představením sítě HKfree a její analýzou na základě teoretických východisek. Důvodem zpracování tohoto tématu jsou jisté nedostatky v oblasti diagnostiky sítě, monitoringu sítě a informovanosti koncových uživatelů o aktuálním stavu. Hlavní náplní tedy bude navržení a vytvoření funkčního diagnostického aparátu, který bude řešit problémy vyplívající z analýzy současného stavu.

Za zrodem myšlenky zpracovat toto téma stojí autorova několikaletá aktivní činnost na pozici správce oblasti v síti HKfree, která zajišťuje dostatečný přehled o tíživé situaci v této oblasti. Zároveň se autorovi dostává reálného prostředí pro testování vzniklého řešení, které bude nadále použito do ostrého provozu.

3 Cíl práce

Cílem této práce je přiblížit teoreticky co je to síť a obecné principy fungování počítačové sítě. Dále pak představit komunitní síť HKfree a na základě teoretických informací analyzovat současný stav této sítě v oblasti monitoringu, diagnostiky a informovanosti o výpadcích. Vzniklá analýza bude použita pro stanovení nedostatků a vytvoření návrhu na zlepšení situace.

Praktická část práce se bude zabývat návrhem a vytvořením řešení ve formě aplikace pro koncové uživatele včetně její serverové podpůrné části, která zjednoduší diagnostiku problémů a zlepší informovanost.

Na závěr bude výsledné řešení testováno, zda opravdu splnilo stanovené předpoklady a přineslo užitek jak koncovým uživatelům, tak pracovníkům technické podpory.

4 Teorie počítačových sítí

Pro bližší orientaci v diplomové práci a seznámení se se zkoumanou problematikou je potřeba nejdříve zevrubně nastínit principy počítačových sítí. Tato kapitola se věnuje definici, rozdělení, použitým prvkům a dalším otázkám způsobu fungování obecně.

4.1 Počítačové sítě

Počítačové sítě obecně mohou mít více způsobů vysvětlení, jeden z nich může být následující: *„Po formální stránce je počítačová síť skupina počítačů popř. periferií, které jsou mezi sebou propojeny tak, aby zajistily vzájemnou komunikaci libovolného uživatele s programem na libovolném počítači, dvou programů mezi sebou nebo dvou libovolných uživatelů mezi sebou, a to při vysoké spolehlivosti komunikace.“* [10] Pro tuto práci ale mnohem lépe charakterizuje počítačové sítě výklad: *“Počítačová síť je systém, který vznikne vzájemným propojením počítačů s cílem komunikovat a společně využívat prostředky připojené k jednotlivým počítačům.”* [4]

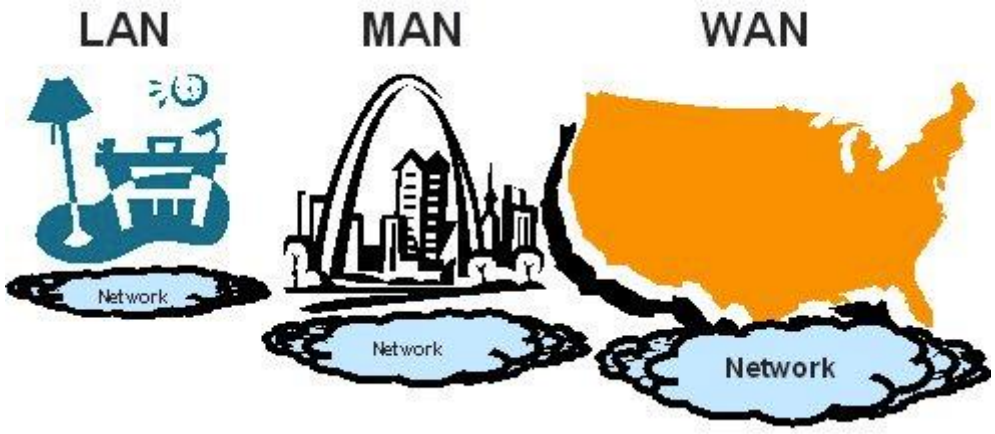
Důvodů vzniku počítačových sítí je v dnešní době mnoho, Horák a Keršláger [3] např. vznášejí otázky na zajištění neustálé aktuálnosti dat: „Jakým způsobem zajistit, aby data byla neustále aktuální? Jakým způsobem přenést data z jednoho počítače na druhý v momentě, kdy se soubor nevejde na datové médium? Jak tisknout na tiskárně, která je připojená k jinému počítači?“ Koutná a Sochor [4] uvádějí další důvody pro vznik počítačových sítí:

- Potřeba společného přístupu k uloženým datům z důvodu nutnosti provádění operací vzdáleně (ukládání, změna, mazání, kalkulace…)
- Přenos dat mezi počítači
- Potřeba společného přístupu k tiskárnám (sdílení nákladných technologií, jednotná údržba…)

4.2 Rozdělení počítačových sítí

Rozdělení počítačových sítí se utvářelo postupně na základě jejich vzniku a následného rozvoje. Složitost sítí se prohlubovala, čítala více a více prvků. Jedním

z několika způsobů dělení sítí je dělení dle velikosti. Následující obrázek 1 graficky znázorňuje velikosti jednotlivých sítí. Jejich charakteristika bude rozebrána níže.



Obrázek 1 Rozdělení sítí dle velikosti
Zdroj: Colégio Liceal de Santa Maria

Typy sítí dle velikosti se dělí na 3 druhy: LAN, WAN a MAN. Příhoda [10] vykládá rozdělení a charakteristiku následujícím způsobem:

Lokální počítačová síť LAN (Local Area Network): Síťové prvky jsou rozmístěny v určitém ohraničeném objektu o rozloze v rozmezí maximálně několika stovek metrů, např.: učebny, školy, městské budovy, firmy atd. Logická i fyzická správa je prováděna pomocí jedné osoby nazývané supervisor nebo administrátor. V případech, kdy lokální sítě nabývají větších rozměrů, může síť spravovat více správců, nicméně pořád musí tvořit koordinovaný tým. Základní skladební prvky jsou většinou osobních počítače a jejich hardwarové prostředky (síťové adaptéry, konektory) spojené síťovými kabely. Přenos dat je realizován pomocí běžně využívaných přenosových médií:

- kroucené dvoulinky
- koaxiální kabely
- vysokorychlostní optické kabely
- bezdrátové spoje

Rozlehlá počítačová síť WAN (Wide Area Network): Prvky rozlehlé počítačové sítě jsou rozmístěny ve více městech, státech i kontinentech. Příhoda [10] zde tvrdí: *„Můžeme bez nadsázky říct, že velikost sítí WAN je dnes omezena velikostí Země.“* Základem sítí WAN jsou tzv. komunikační podsítě tvořené speciálními datovými spoji organizací, poskytujících telekomunikační služby. Tyto datové spoje jsou realizovány pevnými telefonními linkami nebo optickými kabely, výjimkou nejsou ani mikrovlnná a družicová spojení. Komunikační podsítě jsou tvořeny řídicími počítači (tzv. uzlovými počítači, anglicky host). Tyto počítače jsou obvykle velmi výkonné, schopné sloužit většímu počtu uživatelů současně a pracující nepřetržitě. Za uzly WAN se považují i jednotlivé LAN, které mezi sebou komunikují právě prostřednictvím rozlehlé sítě. Vzájemné propojení více počítačů probíhá zprostředkovaně, protože u rozlehlých sítí je není možné propojit přímo. Data jsou předávána postupně od jednoho počítače k druhému, a to až k cílovému počítači.

Jako příklad takové sítě uvádí autor síť českých univerzit a vědeckých institucí CESNET2 a samozřejmě největší světová síť Internet.

Městská počítačová sít MAN (Metropolitan Area Network): Jednotlivé počítače jsou rozmístěny v rozsahu města, v řádech několika kilometrů. Díky zlepšování komunikačních prostředků jsou tyto sítě stále více podobné LAN sítím, z čehož vyplývají i jejich způsob použití. Na rozdíl od LAN sítí ale ke spojení využívají i veřejné komunikační sítě.

Jako příklad takové sítě uvádí autor síť Univerzity Palackého v Olomouci – UPONET, která se nachází na celé rozloze města Olomouc.

S tímto rozdělením sítí se ztotožňují i autoři Horák a Keršláger [3].

Počítačová síť provozovaná spolkem HKfree je klasifikována jako lokální počítačová síť MAN i přes její značný rozsah a geografické pole působnosti. Proto bude nadále používáno označení „síť“ ve smyslu lokální počítačové sítě MAN.

V následující tabulce jsou pro porovnání uvedeny služby zajišťované lokálními a rozlehlými počítačovými sítěmi.

LAN	WAN
sdílení periferií (laserové tiskárny, velkokapacitní diskové systémy, apod.)	práce na vzdálených počítačích (remote login)
sdílení společných dat a aplikací (aktuálnost dat, úspora diskového prostoru, snadné zálohování, apod.)	přenos dat (FTP), elektronická pošta (e-mail)
využívání Intranetu a jednoduchou komunikaci mezi uživateli	přístup do rozsáhlých informačních databází, konference, diskusní kluby
	WWW (World Wide Web)

Tabulka 1 Služby zajišťované sítěmi
Zdroj: Autor

Dalším již méně používaným způsobem rozdělení může být rozdělení dle topologie sítě, tím se rozumí způsob zapojení prvků do funkčního celku.

Sběrnicová topologie: způsob zapojení počítačů do sítě pomocí jednoho jediného prvku, tzv. sběrnice. Výhodou jsou malé pořizovací náklady a jednoduchost zapojení. Nevýhodou je potom možnost komunikace pouze jednoho zařízení naráz, nebo dojde ke kolizi. S rostoucím počtem zapojených počítačů roste riziko kolizí a musí být přijímána ochranná opatření.

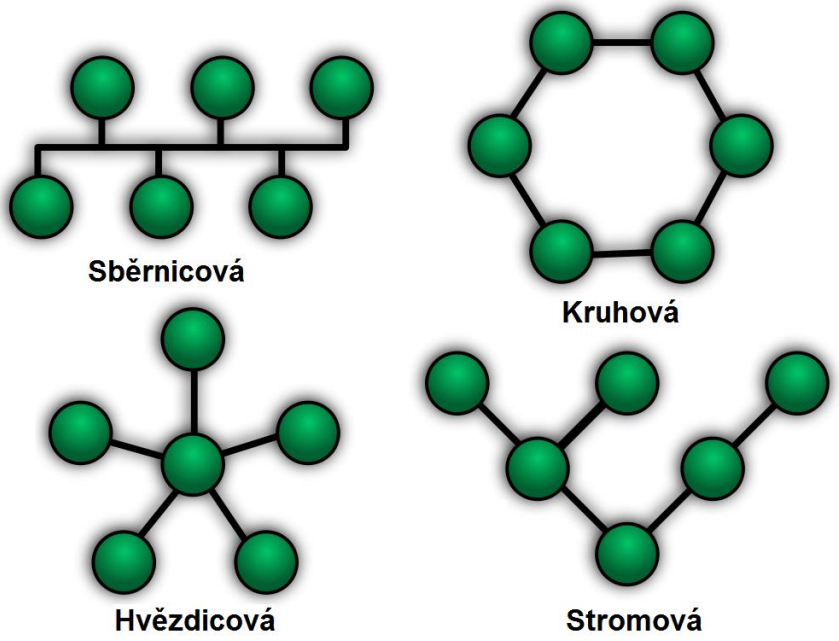
Kruhová topologie: je způsob zapojení jednotlivých stanic do kruhového uspořádání. Přenos informací probíhá na základě vyslání tokenu, který putuje skrz jednotlivá zařízení v kruhu až na místo určení s tím, že když si ho stanice přečte a není pro ni určen, ihned ho odesílá dál. Nevýhoda tohoto zapojení je v tom, že při přerušení kruhu může dojít k nefunkčnosti celé sítě. Pro předejití takovým situacím se používají záložní okruhy.

Hvězdicová topologie: nejvyužívanější způsob propojení počítačů do sítě v dnešní době. Počítače jsou připojeny pomocí kabelu k centrálnímu prvku (přepínač, rozbočovač), který zajišťuje jejich propojení. Pokud selže kabel nebo jeden počítač,

ostatní část sítě funguje dál pouze bez dostupnosti toho počítače. Pokud selže centrální prvek, bude nedostupná celá část sítě spravovaná tímto prvkem. Tento druh propojení se hodí pro vysokorychlostní přenosy a má nižší nároky na instalovanou kabeláž.

Stromová topologie: vychází z hvězdicové topologie zapojení sítě a funguje na principech pospojování jejich centrálních prvků do obrazce připomínajícího strom. Výhodou je snižování nároků na kabeláž a eliminace celkových výpadků sítě.

S tímto způsobem dělení dle topologie se ve své knize ztotožňuje Bigelow [1] i Horák a Keršláger [3].



Obrázek 2 Rozdělení sítí dle topologie
Zdroj: Autore

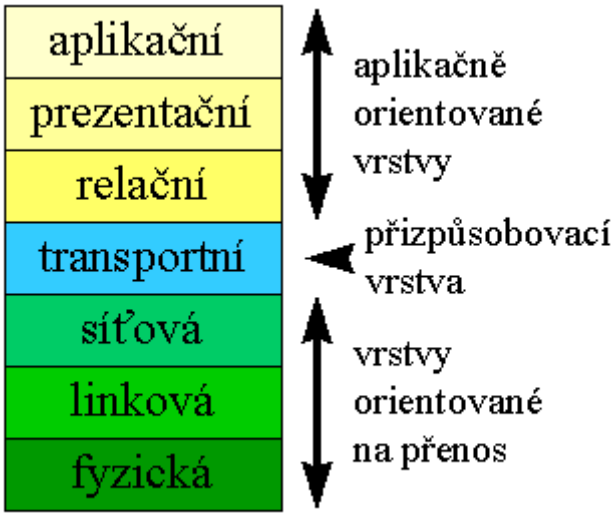
Obrázek 2 ilustruje rozdělení sítí dle topologie.

4.3 Model ISO/OSI

Po obecném představení sítě je potřeba přiblížit na jakých principech funguje. K tomu poslouží Referenční model ISO/OSI, který zpracovala organizace ISO (světová federace národních normalizačních organizací se sídlem v Ženevě) za

účelem standardizace počítačových sítí zvané OSI (Open Systems Interconnection). V roce 1984 byl přijat jako mezinárodní norma ISO 7498.

Komunikaci mezi dvěma body v síti řídí síťové protokoly. Tento model charakterizuje přenos dat v sítích a mezi nimi, rozděluje síťovou architekturu do 7 vrstev a popisuje vztahy mezi nimi. Protokol určité vrstvy pak nabízí své služby vrstvám vyšším a využívá služby vrstev nižších.



Obrázek 3 Přehled vrstev referenčního modelu ISO/OSI
Zdroj: www.earchiv.cz

Výklad jednotlivých vrstev se může mírně lišit, pro účely této práce je upřednostněno pojetí Horáka a Keršlágera [3], které je následující:

Aplikační vrstva (Application Layer) je aplikací zprostředkovávající uživatelům síťové služby. Mezi takové funkce patří vzdálený přístup k tiskárnám, elektronické zprávy, přístup k souborům na vzdálených počítačích atp.

Prezentační vrstva (Presentation Layer) má na starosti konverzi různě kódovaných dat z různých sítí a zajišťuje sjednocení formy vzájemně přenášených údajů. Dále se stará o jejich komprimaci a případné šifrování.

Relační vrstva (Session Layer) spojuje a po dokončení přenosu ukončuje vytvořené spojení. Dále má na starosti bezpečnost vytvářeného spojení a může provádět ověřování uživatelů nebo zabezpečení přístupu k zařízením.

Transportní vrstva (Transport Layer) se stará o přenos zpráv, ty dělí na pakety a opět zase skládá dohromady do zpráv. Během přenosu se mohou pakety pomíchat či ztratit.

Síťová vrstva (Network Layer) spojuje a směruje tok paketů mezi dvěma počítači nebo celými sítěmi, mezi nimiž není přímé spojení. Volí trasu spojení (mezi sítěmi bývá více možných cest pro přenos paketů), volba trasy je nazývána směrováním – routingem.

Linková vrstva (Data – link Layer) zajišťuje přenos datových rámců po fyzickém médiu pomocí fyzických adres síťových karet, odesílá a přijímá rámce a prověřuje cílové adresy všech přijatých rámců. Dále určuje, zda bude rámec odevzdán vyšší vrstvě nebo nikoliv.

Fyzická vrstva (Physical Layer) popisuje elektrické, optické, mechanické a funkční vlastnosti:

- jaký signál představuje logickou jedničku
- jak stanice rozezná začátek bitu
- jaký je použit konektor
- využití vodičů v kabelu atd.

V praktické části bude vysvětlena problematika síťového provozu a jeho směrování z důvodu diagnostiky, proto je ještě více potřeba přiblížit síťovou vrstvu modelu.

„Pokud komunikující systémy sdílí společný segment sítě LAN, mohou být pakety vyměňovány přímo pomocí datových a fyzických vrstev zdrojového i cílového

systému. Pokud jsou však zdrojové a cílové systémy v různých sítích, musí směrovače přesouvat pakety v síti trasou, která je již nadefinována nebo je dynamicky zjištěna.“ [1]

„Síťová vrstva rozhoduje o tom, jakou cestou budou postupně přenášena data, která se mají dostat k určitému konkrétnímu adresátovi. Zformuje zprávu z transportní vrstvy do datových paketů, které pak mohou nižší dvě vrstvy přenášet.“ [4]

4.4 Síťové prvky

Aby počítačová síť mohla fungovat a vůbec existovat, musí se skládat z určitých částí. Tyto části jsou nazývány síťové prvky. Jejich rozdělení je dle Horáka a Keršlágera [3] následující:

- **Síťové počítače:** běžná PC pracující v síti
- **Síťový hardware:** síťové karty v počítačích, kabely, aktivní prvky v kabeláži
- **Síťový software**: programy na síťových stanicích, případně serverech

Toto rozdělení je ale pro potřeby této práce málo podrobné a mohlo by vést k nedostatečné informovanosti o základních stavebních prvcích sítě, používaných pojmů a celkovému nepochopení, jaký význam a způsob fungování bude mít výsledné řešení. Proto bude v této práci vycházeno z podrobnějšího rozdělení, sestaveného dle seznamu běžně používaných síťových prvků vyplývajících z předchozích kategorií.

- Opakovač (Repeater)
- Rozbočovač (Hub)
- Most (Bridge)
- Směrovač (Router)
- Přepínač (Switch)
- Brána (Gateway)

Některé prvky jsou již v dnešní době zastaralé a dávno nahrazené modernějšími technologiemi, proto budou dále popsány pouze ty z nich, které se v HKfree běžně využívají. Příhoda [10] vykládá jejich popis tímto způsobem:

Přepínač (switch) pracuje na fyzické vrstvě a jeho pracovní úloha spočívá v paralelní komunikaci mezi uzly různých segmentů sítě. Příhoda [10] přirovnává funkci k telefonní ústředně. Do budovy vede několik linek a uvnitř je několik koncových stanic. Přepínač propojuje linku, která vede zvenku s koncovým uzlem uvnitř nebo naopak dle svých pravidel. To účastníky komunikace netrápí a ani je to nezajímá, potřebují pouze spojení.

Směrovač (router) operuje na síťové vrstvě a spojuje různé segmenty sítě. Principem jeho činnosti je nalezení cesty v síti - směrování. Jeho počínání v síti není transparentní, ale stanice o něm ví. Nalezne uplatnění tam, kde sítě mají různé adresy, pro připojení k Internetu nebo pro komunikaci mezi sebou. Směrovače musí mezi sebou pracovat se stejným protokolem, toto na fyzické a linkové vrstvě platit nemusí.

Brána (gateway) je zařízení zajišťující propojení heterogenních sítí. Pracuje na vrstvě, kde může bezpečně proběhnout konverze protokolů. To může být až vrstva aplikační. V dnešní době se již většinou používají zařízení, která obsahují funkčnost brány a směrovače zároveň. Příkladem takového zařízení je běžný domácí přístupový bod.

Jedná se o popis spíše laického charakteru, ale práce nevyžaduje pochopení hlubšího způsobu principu fungování těchto prvků.

4.5 Směrovací protokoly

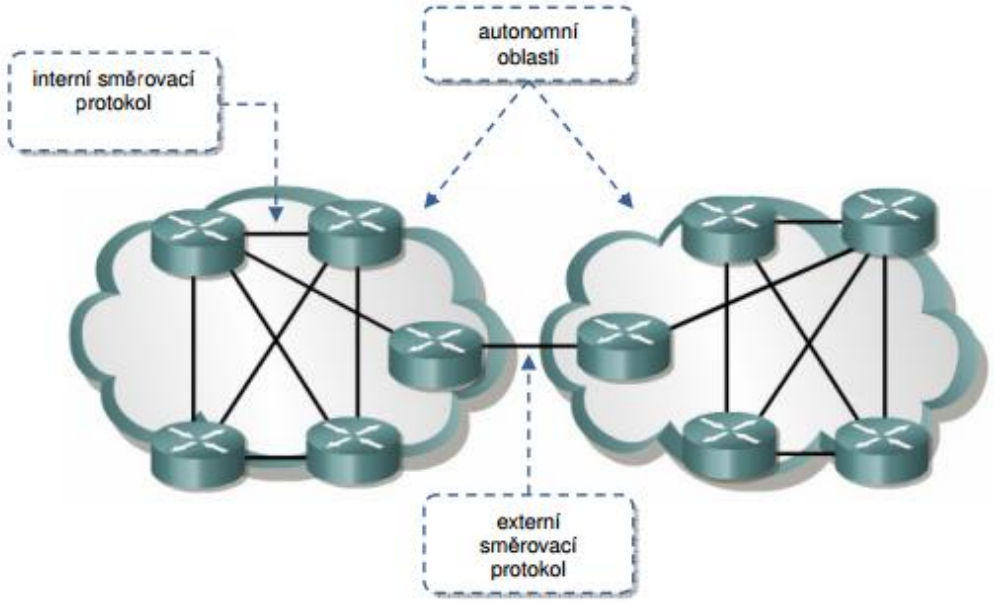
Již byla představena síť, z čeho se skládá a na jakých principech se přenášejí data z jednoho zařízení na druhé. To, jakou cestou se tam data dostanou, řeší tzv. směrovací protokoly. V předchozích kapitolách bylo řečeno, že směrovače posílají

pakety cestou, která je definována nebo dynamicky zjišťována. Tyto způsoby řeší ve své práci Lomnický a Veselý [7], kteří pojednávají o dvou způsobech směrování:

1) **Statické** – směrování na základě statických záznamů ve směrovací tabulce, jinými slovy směrování na adresy, které administrátor sám ručně do směrovací tabulky zadá.
2) **Dynamické** – řídí se dynamickými směrovacími protokoly, které na základě určitých algoritmů vyhodnocují optimální trasy v měnící se síti.

Dynamické směrovací protokoly se dále ještě dělí na interní a externí. Interní se používají na směrování dat v uzavřené síti nebo autonomním systému (AS), který má většinou více než jeden možný přístup do ostatních externích systémů. Takové sítě většinou lze přirovnat k poskytovatelům internetu, jejichž zákazníci patří do jejich autonomního systému. Příklad takových protokolů je RIP (Routing Information Protocol) a OSPF (Open Shortes Path First protocol). Externí směrovací protokoly pak zajišťují směrování mezi těmito autonomními systémy a externími sítěmi. Příklad takového protokolu je BGP (Border Gateway Protocol).

Na následujícím obrázku 4 je ilustrováno, jaké směrovací protokoly se používají v různých částech sítě. V tomto konkrétním případě je možné představit si autonomní systém jako síť HKfree a její plzeňský protějšek Pilsfree. V obou případech bude pro interní směrování použit protokol OSPF. Mezi těmito autonomními systémy pak bude použit externí směrovací protokol BGP.



Obrázek 4 Interní a externí směrovací protokoly
Zdroj: Směrování a směrovací protokoly

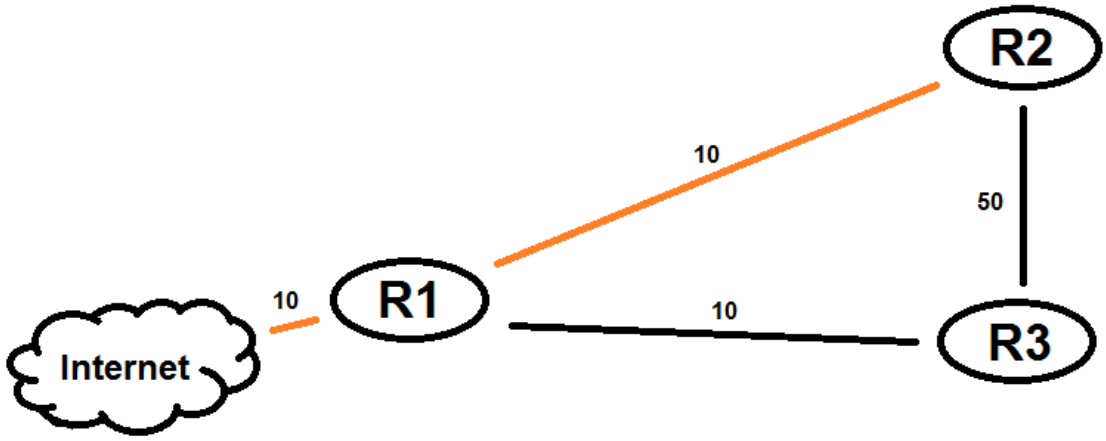
V síti HKfree se již v tuto dobu statické směrování téměř nenachází a používá se právě směrování OSPF. Jeho funkčnost je pro potřeby sítě zcela vyhovující, což ostatně uvádí i Bigelow [1] ve svém tvrzení: *„Použití protokolu OSPF je v mnoha ohledech výhodnější než použití protokolu RIP, jeho silné stránky se však projevují zvláště v rozsáhlých síťových prostředích, ve kterých rychlé a efektivní aktualizace tras velmi přispívají ke stabilitě sítí při změnách ve směrování.“* Toto tvrzení potvrzuje i článek publikovaný společností Microsoft [9]: *„Největší výhodou protokolu OSPF je jeho efektivita - tento protokol ani ve velmi rozsáhlých sítích nepředstavuje nijak významné zatížení přenosových cest.“*

Dále společnost Microsoft [9] uvádí definici protokolu OSPF, která pro tuto práci zcela vyhovuje a dostatečně popisuje funkčnost protokolu: *„Protokol OSPF počítá trasy ukládané do směrovací tabulky pomocí algoritmu SPF (Shortest Path First). Tento algoritmus hledá nejkratší (nejméně nákladnou) cestu mezi směrovačem a jednotlivými dostupnými sítěmi.“*

Směrovače v sobě uchovávají vypočítané mapy s nejoptimálnějšími trasami, které se nazývají databáze stavů linky (Link State Database). Tyto databáze jsou

aktualizovány při každé změně síťové topologie a při jejich výpočtu jsou uplatňována ochranná opatření proti vytvoření smyčky. Takto vytvořené mapy jsou efektivně přenášeny mezi jednotlivými směrovači tak, aby vždy a všude byly aktuální hodnoty.

Na následujícím obrázku 5 je ilustrován postup výpočtu optimální trasy protokolem SFP. Každá trasa je ohodnocena nákladností (cost) a je použita ta nejméně nákladná k cíli. V tomto případě paket putující ze směrovač R2 použije přímou cestu přes směrovač R1, protože cost cesty má hodnotu 20. Pokud by na trase R2 - R1 došlo k výpadku, protokol cestu přepočítá, vytvoří novou mapu, tu rozdistribuuje do všech směrovačů a pošle paket druhou nejnižší možnou cestou, což bude přes směrovač R3 s costem 70.



Obrázek 5 Optimální trasa OSPF
Zdroj: Autor

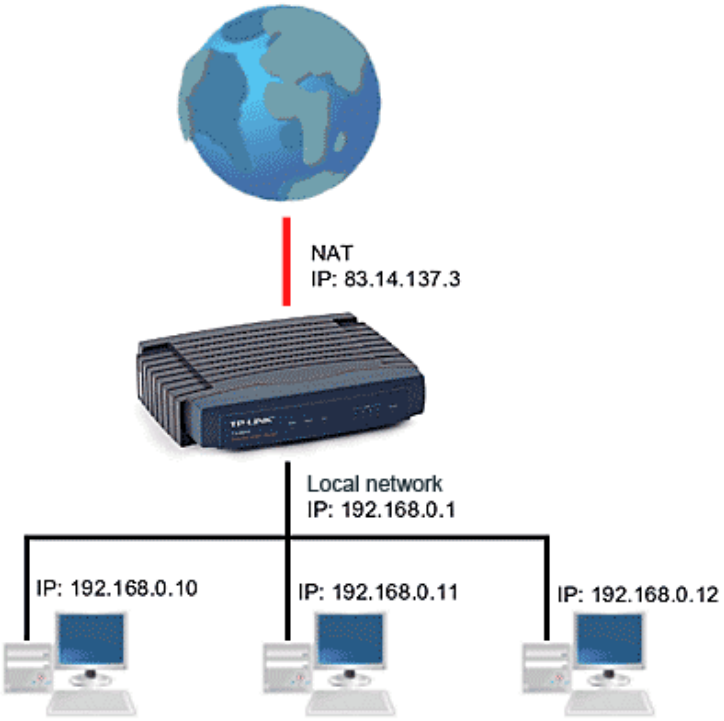
4.6 NAT

Představen musí být v teoretické části i NAT (Network Address Translation), který je v domácnostech hojně využíván. *„Jedná se o funkci routerů, která umožňuje překládat adresy z vnitřního adresního rozsahu do veřejného a naopak. V důsledku se tak vnitřní adresy nedostanou nikdy do internetu.“* Charakterizuje ve svém článku NAT Krčmář [5] a dále uvádí skutečnost, že vznik NATu je důsledek malého počtu veřejných IP adres. Ne každý může používat adresy z vnějších rozsahů, a proto byl

vymyšlen princip „schovávání“ celých vnitřních rozsahů za jednu adresu. Podle Krčmáře [5] funguje překládání adres na těchto principech:

- Klientem jsou vyslány pakty na bránu vnitřní sítě.
- Pakety jsou zachyceny směrovačem a do jejich hlavičky je vložena vnější IP adresa.
- Pakety jsou směrovačem odeslány z náhodného portu TCP.
- Směrovač si uchová informace o tom, kterým portem data odeslal a jakému zdrojovému klientovi patří.
- Po příchodu odpovědi směrovač dle uložených informací přepošle data zdrojovému klientovi.

Tuto skutečnost potvrzuje Bigelow [1] a dokládá ve své knize stejným postupem zasílání paketu skrz NAT. Následující obrázek 6 ilustruje schéma fungování, kde jsou místní počítače v interním adresním prostoru schovány za jednou adresou z externího adresního prostoru.



Obrázek 6 Princip fungování NAT
Zdroj: www.dipolnet.com

Z tohoto postupu je patrné, že pokud komunikaci nenaváže sám klient schovaný ve vnitřní síti za NATem, je nemožné ho jakkoliv kontaktovat a navázat s ním spojení. Pro účely této práce a vývoj diagnostického aparátu, který by měl, jako jednu z vlastností, obsahovat komunikaci mezi samotnými klienty, toto představuje problém, který bude potřeba vyřešit.

Existují postupy, které dokáží i přes to sestavit spojení mezi klienty v „zanatovaných" sítích. Jedením z nich je nat-traverse, který pracuje způsobem zasílání požadavků pomocí UDP (User Datagram Protocol) protokolu. UDP je bezstavový protokol, po jehož odeslání neexistuje kontrola správného doručení. Krčmář [6] popisuje funkčnost nat-traverse následujícím způsobem:

- PC1 odešle náhodná data pomocí UDP paketu na adresu a port NAT2 sítě, která neví o žádné komunikaci, a pakety zahodí.
- PC2 odešle náhodná data pomocí UDP na adresu a port NAT1 sítě, ta si již myslí, že se jedná o odpověď na první pakety, protože přicházejí od NAT2 a proto je NAT1 doručí až k PC1.
- PC1 znovu pošle náhodná data pro NAT2 a nyní si i NAT2 myslí, že se jedná o odpověď, a paket předá k PC2.

Jakmile dorazí pakety k oběma PC, NAT si již myslí, že se jedná o normální komunikaci a sestaví UDP most mezi oběma počítači. Ten lze poté využít pro sestavení TCP tunelu.

4.7 Nástroje pro řešení potíží

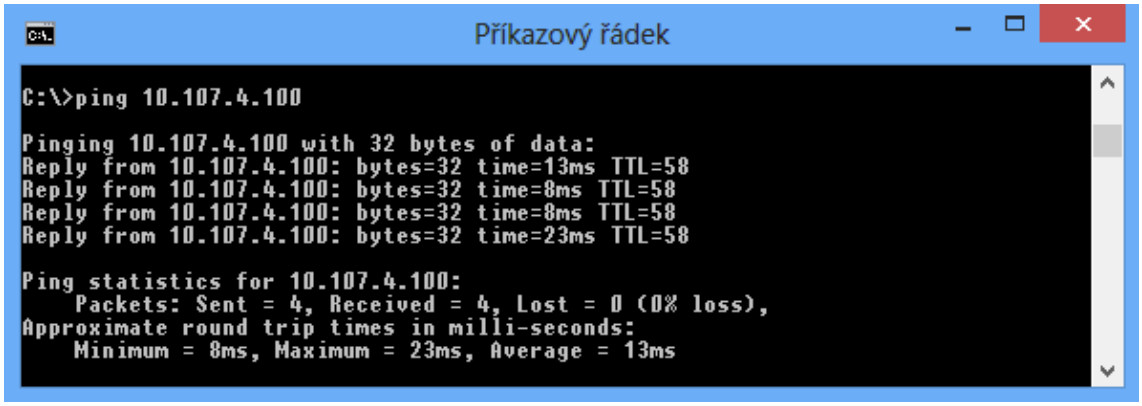
Protože předmětem diplomové práce je právě diagnostika a řešení potíží s připojením, je nutno uvést nástroje, které budou v praktické části použity. Jedná se o již zabudované součásti většiny systémů a není třeba je dodatečně instalovat nebo přenášet.

4.7.1 Ping

Longer [8] uvádí ve svém článku na portálu pcblog.cz definici: *„Příkaz ping je jedním z nejjednodušších příkazů, které používáme pro diagnostiku problémů na síti.*

Jeho návratovou hodnotou je čas, který potřebuje paket k cestě z vašeho počítače k cílovému počítači a zpět (tzv. roundtrip time). Tento čas je obvykle udáván v milisekundách.“, která naprosto vystihuje základní funkčnost příkazu ping. Z tohoto tvrzení vyplývá, že pokud má být zjištěna dostupnost cílového počítače v síti, lze toho docílit použitím příkazu ping se vstupní hodnotou síťové adresy cílového počítače. Dle návratové hodnoty je možné zjistit, zda je počítač dostupný a pokud ano, s jakou přístupovou dobou. Těmito údaji lze provádět diagnostiku. Předchozí informace o příkazu ping potvrzuje ve své knize Bigelow [1] a dodává, že Ping: *„je nepostradatelným nástrojem pro řešení potíží s konektivitou sítě Internet a místní sítě.“*

Dále Bigelow [1] podrobněji popisuje fungování Pingu následujícím způsobem. Ping odesílá zprávy Echo ICMP (Internet Control Message Protocol) cílovému počítači. Pokud je síť v pořádku a správně nakonfigurovaná, přijme cílový počítač všechny zprávy Echo ICMP a na každou z nich odpoví zprávou Echo Response, která potvrzuje všechna přijímaná data. V případě, že počítač odesílající příkaz Ping obdrží všechny zprávy v předem stanoveném čase zpět, znamená to, že síť mezi těmito počítači je správně nakonfigurována a funkční pro přenos dat. Na obrázku 7 lze vidět, jak vypadá Ping na fungující hlavní počítač HKfree.



Obrázek 7 Ping na hlavní počítač HKfree
Zdroj: Autor

Autoři se shodují, že v ojedinělých případech může být použití Ping nepřesné, protože některé síťové prvky (například od výrobce CISCO) zpracovávají pakety

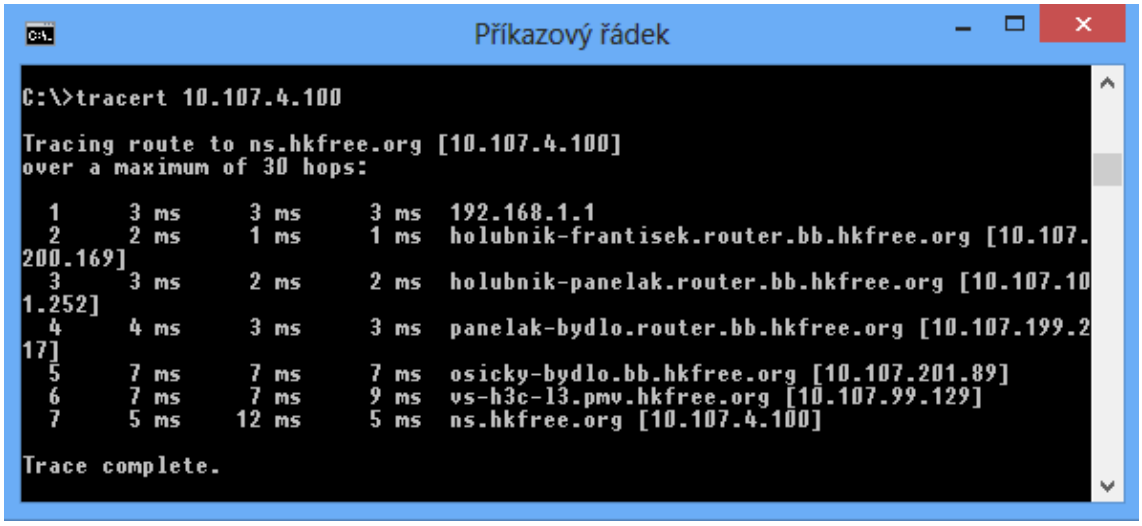
ICPM pro ně určené s poměrně malou prioritou a následkem je žádná nebo déle trvající odpověď. Děje se tomu tak hlavně z důvodu upřednostnění důležitějšího síťového provozu, jako je například HTTP protokol (Hypertext Transfer Protocol) zajišťující přenos internetových stránek ve formátu. Podobný problém způsobují i brány Firewall, které mohou blokovat ICMP pakety.

4.7.2 Traceroute

Jedná se diagnostický program pro platformu Windows (Tracert), který dokáže vypsat trasu paketu podstoupenou putováním sítí. V praxi se používá na diagnostiku sítě, zda je správně nastaveno směrování, případně kde je v síti problém.

Dostálek [2] popisuje jeho funkčnost tak, že využívá protokolu ICMP k odesílání Echo paketů, kterým nastaví jako první hodnotu TTL (Time To Live). Tuto hodnotu musí každý směrovač snížit o jeden stupeň při jeho průchodu. První Echo paket je odeslán s hodnotou 1 a hned první směrovač vrátí zpět paket „čas vypršel“, protože snížil hodnotu na 0. Traceroute zaznamená potřebnou dobu k reakci a adresu posledního hostitele ze záhlaví IP. Celkem provádí tyto pokusy 3x s jedním nastaveným TTL, potom pokračuje tím, že zvedne TTL o jednotku a paket se dostane dál. Cyklus končí ve chvíli, kdy je TTL paketu dostatečně velké a dostane se k cílovému počítači, který odpoví, že byl doručen. Výsledné hodnoty zapisuje Traceroute do tabulky, kterou je možné vidět na konci kapitoly.

Trasování může být taky ukončeno ve chvíli, kdy směrovač již nezná cestu dál a odpoví „nedoručitelný paket“. V tomto případě je známa adresa posledního směrovače, která je zásadní pro diagnostiku sítě.



Obrázek 8 Traceroute na hlavní počítač HKfree
Zdroj: Autor

5 Analýza prostředí a stavu

Na základě východisek, která byla představena v předešlých kapitolách, se práce bude v praktické části věnovat představení sítě HKfree, otázkám které směřují k sestavení analýzy současného stavu sítě, postupů diagnostiky a následnému návrhu na zlepšení situace a návrhu diagnostického systému.

5.1 HKfree

Na začátku všeho stála pouze myšlenka, kterou sdílel velmi omezený počet nadšenců. Tito lidé se rozhodli sdílet, v tu dobu, ještě velice nákladné a pomalé připojení k internetu a dostat se tak informacím. Nároky neustále rostly a technologie prožívaly bouřlivý vývoj. To, co bylo dříve finančně nedosažitelné, náhle dostávalo jasnějších rozměrů a počet lidí, zajímajících se o informační technologie obecně, stále rostl. Tomuto uskupení bylo potřeba dát nějakou formu a tak vzniklo Občanské sdružení HKfree, v současné době spolek hkfree.org, dále jen HKfree.

Jeho název vzniká spojením slov „HK“ dle mateřského města Hradce Králové a „free“ jakožto určitým vyjádřením volnosti a svobody. Již od samého začátku funguje spolek na principu dobrovolnické činnosti a každý člen vykonává svoji aktivitu na základě vlastní ochoty, a bez nároku na jakoukoliv mzdu. Hnacím motorem je pro tyto příznivce jednodušší přístup k nejnovějším technologiím a možnost si vše osahat na vlastní kůži. Protože je tento koníček velice nákladný, vybírají se od všech členů členské příspěvky, které vytvářejí společné finance a jelikož se jedná o neziskovou organizaci, tak veškeré tyto finance putují pouze na provoz a inovace sítě. Přebytečné finance jsou pak použity na financování různých veřejně prospěšných projektů.

Jako každý oficiální subjekt musí mít i HKfree svojí danou strukturu. Nejvyšším orgánem je Valná hromada, ta řeší základní otázky fungování spolku a finanční toky, shází se jednou ročně. Mezi těmito obdobími se stará o chod Výkonný výbor čítající 5 členů, který je volen jednou za rok Valnou hromadou. V posledních letech

byl pro podpůrné aktivity sestaven kontrolní orgán s názvem Komise pro rozvoj, jejímž cílem je přezkoumávat rozhodnutí Výkonného výboru a řešit otázky s menší mírou zodpovědnosti. Na venek zastupují spolek dva zvolení Jednatelé, voleni Výkonným výborem, též na jeden funkční rok.

Na nejnižší správní úrovni jsou správci oblastí, tvoří základní kameny spolku a svojí dobrovolnou činností spravují, budují a udržují jednotlivé části, dnes již velice rozlehlé sítě, která pokrývá celou rozlohu Hradce Králové a přilehlé obce až do vzdálenosti 30 kilometrů. Jejich neustále se proměňující počet čítá kolem 50ti jedinců. Správcem oblasti, dále jen SO, se člověk stane v případě, že se sám zasadí o vybrání a zprostředkování vhodného umístění pro instalaci technologií, jejich následné instalaci a správě. SO dále slouží jako styčná osoba pro zájemce o připojení do sítě a je jim s tímto úkonem nápomocen.

Z důvodu narůstajícího množství zájemců o připojení do sítě, které poslední dobou převyšovalo volnočasový rozpočet správců, musela být zřízena externí technická podpora. Pracovníci technické podpory, dále jen technici, jsou nyní jediná placená pracovní síla celého spolku, čítajícího přes 4000 členů, hrazená z rozpočtu. Jejich náplní práce je připojování nových uživatelů, podpora pro stávající uživatele, inovace sítě a opravy stávající infrastruktury. To vše vykonávají pouze za předpokladu, že místní SO nemá na dané úkony čas, nebo na ně sám nestačí. Vše musí být před zásahem odsouhlaseno Výkonným výborem, včetně finanční a časové dotace tak, aby byla co nejvíce zachována původní myšlenka komunitní sítě, spravované pouze jejími členy zdarma, a ve volném čase.

Každý člen spolku má, nebo může mít přímo úměrně své míře a chuti, zapojení do chodu sítě přístup k různým technologiím. HKfree vždy podporovalo a bude podporovat vzdělání svých členů a maximálně je v tomto ohledu podpoří. Přijde-li člen se smysluplnou a obhajitelnou myšlenkou na projekt zabývající se informačními technologiemi, jen zřídka se nesetká s plnou podporou spolku. Všechny tyto aspekty dělají z HKfree něco více než jen zprostředkovatele připojení k Internetu.

5.2 Síťová infrastruktura

Celou počítačovou síť HKfree lze prezentovat jako autonomní systém, který má více než jeden přístup do okolních externích systémů. Tzv. „mozek“ tvoří serverovna, umístěná v pronajatých prostorách administrativní budovy, která leží na přípojném bodě optické trasy. Zde je realizována hlavní konektivita do sítě Internet a ostatních sítí za pomoci brány (gateway). Tato brána je umístěna na výkonném serverovém počítači, který integruje zároveň služby směrování (routing) a filtry, dle kterých je rozhodováno, které vnitřní počítače mají či nemají přístup do Internetu. Celé toto zařízení je v dalších částech práce prezentováno jako hlavní počítač HKfree z důvodu srozumitelnějšího výkladu koncovým uživatelům reprezentujícím laickou veřejnost.

Tento centrální bod je na obrázku 9 reprezentován jako červená tečka a zhruba uprostřed sítě je umístěn i geograficky. Dále serverovnu tvoří další přidružená zařízení starající se o chod sítě a hostující používané systémy. V rámci zachování bezpečnosti a integrity dat, jsou používané systémy provozovány na oddělených virtuálních serverech. Jedním z nich je virtuální stroj, hostující monitoringové služby, popsané v další kapitole.

Z centrálního bodu je síť rozváděna pomocí drátových a bezdrátových linek do okolních strategických míst zvaných „přístupové body“. Tento název je používán z překladu AP (Access Point). Přístupové body jsou vždy realizovány za použití směrovače HW, potřebného pro realizaci primárních linek a HW, potřebného pro realizaci konektivity pro koncové uživatele. Směrovače mají v rámci sítě jedinečné nejen adresy, ale i jmenné názvy, které jsou využívány pro identifikaci. Na obrázku 9 je znázorněna mapa všech přístupových bodů sítě, reprezentovaných žlutými tečkami (pořízeno softwarem OSPF-visualiser, [11]).

Protože se jedná o komplikovanou infrastrukturu se spoustou důležitých bodů, je potřeba zajistit určitou úroveň spolehlivosti a dostupnosti při výpadcích způsobených například selháním HW nebo přírodními podmínkami. Opatření jsou

prováděna primárně výstavbou záložních linek, které v dnešní době pokrývají naprostou většinu přístupových bodů. Méně časté výpadky, způsobované přerušením dodávky elektrického proudu lokálního charakteru, jsou do určité míry ošetřeny instalací záložních zdrojů elektrické energie UPS (Uninterruptible Power supply Source).

Pro směrování v rámci sítě je použit směrovací protokol OSPF, který díky svému způsobu fungování nejlépe vyhovuje požadavkům sítě. Zejména se jedná o vypočítávání optimálních tras, měnících se v závislosti na aktuálním dění v rozlehlých oblastech sítě a na základě využívání primárních či záložních linek.

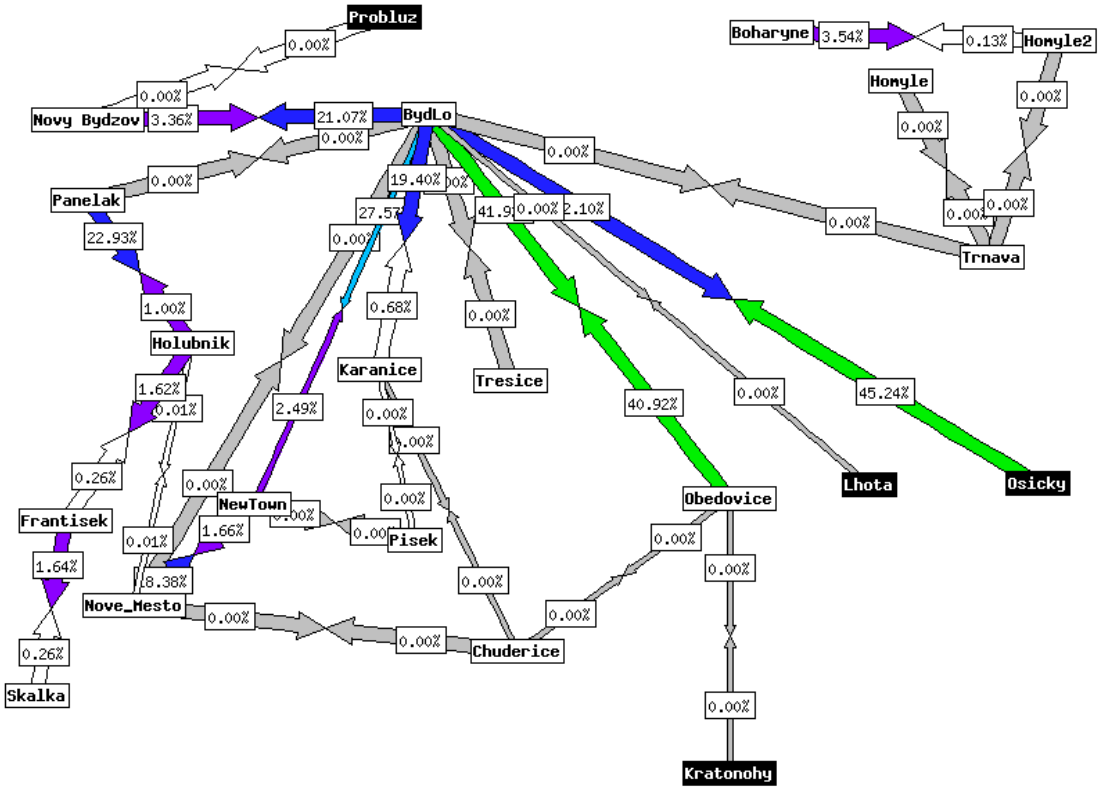
Na obrázku 9 je dále znázorněna struktura sítě, respektive struktura směrovačů tvořících páteřní body sítě a zaštiťujících jednotlivé oblasti.

Obrázek 9 Topologie sítě HKfree
Zdroj: Autor

5.3 Používané monitorovací nástroje

Právní forma spolku a samotná organizační struktura, předurčuje HKfree, na rozdíl od jasně dané hierarchie pracovních postupů a vedení komerčních ISP (Internet Access Provider), spíše k dobrovolnické činnosti většiny zainteresovaných členů a tím pádem, k jistému živelnému postupu při výběru a zpracování dílčích procesů, nevyjímaje ani otázku monitoringu sítě. V současné době existuje několik monitorovacích systémů současně, a každý správce je zodpovědný za zanesení svých spravovaných oblastí – přístupových bodů představených v síťové infrastruktuře do těchto systémů. Situace je komplikována především tím, že tato činnost není povinná, nýbrž dobrovolná a pouze doporučovaná. Jako další úskalí vedení uceleného monitoringu sítě bylo zjištěno vkládání informací do těchto systémů. Neexistuje zde žádný společný mechanismus jednoznačné identifikace síťových prvků (směrovače, přepínače atp.), záznamu vstupních parametrů a následné distribuce do všech systémů, ale každý systém sám má svou odpovědnou osobu, která vkládání dat provádí, a slouží jako kontaktní osoba pro příjem potřebných informací od jednotlivých správců.

Cacti, open-source monitorovací systém, je v současné době nejsilnějším a nejdéle používaným nástrojem spolku. Získané informace zpracovává pomocí RRDTool (nástroj pro zpracování a ukládání časově závislých dat) a následně zobrazuje ve webovém prohlížeči. Je využíván pro sledování síťových prvků a to především vytížení jejich komunikačních portů, měření uptime (doba provozu od spuštění systému), vytížení CPU a celkové dostupnosti. Tyto funkce zastřešuje nástroj „weathermap“, který jednotlivé prvky spojuje v celek, a zobrazuje jako jednotlivé mapy sítě s celkovou dostupností a aktuální propustností. Náhled na reálný provoz je možné vidět na obrázku 10. V praxi pak dochází k podobné vizualizaci sítě jako je na obrázku 9, ale doplněné o údaje, zobrazující aktuální stav linek.



Obrázek 10 Monitoring sítě pomocí Cacti
Zdroj: Autor

Dalším hojně využívaným nástrojem je **SmokePing**. Jednoúčelová a nenáročná webová aplikace sloužící pro záznam dostupnosti síťových prvků na časové ose spolu s jejich latencí. Na základě takto naměřených dat, získaných pomocí nástroje Ping a jeho odpovědí, lze provádět analýzu technického stavu sítě v návaznosti na denní době, a plánovat možné opravy nebo posílení infrastruktury. Použití tohoto nástroje je však ztíženo absencí modulu pro webovou správu a veškeré informace je nutno zanášet přímo do konfiguračního souboru. Tímto téměř odpadá naděje zaznamenávání koncových zařízení a monitoring takzvané poslední míle (= konektivita mezi přístupovým bodem a koncovým uživatelem), který není tímto způsobem udržitelný.

Následující zkoumaný software **Nagios** patří mezi rozsáhlejší s poměrně krátkou dobou používání v HKfree. Agreguje v sobě všechny hlavní funkcionality dříve zmíněných řešení. Jeho hlavní výhodou je zasílání zpráv o aktuálním stavu

síťových prvků a jejich změnách pomocí emailů či jiných komunikačních kanálů. I když se jeví tento nástroj jako nejsilnější, jeho využití je nyní stále minoritní.

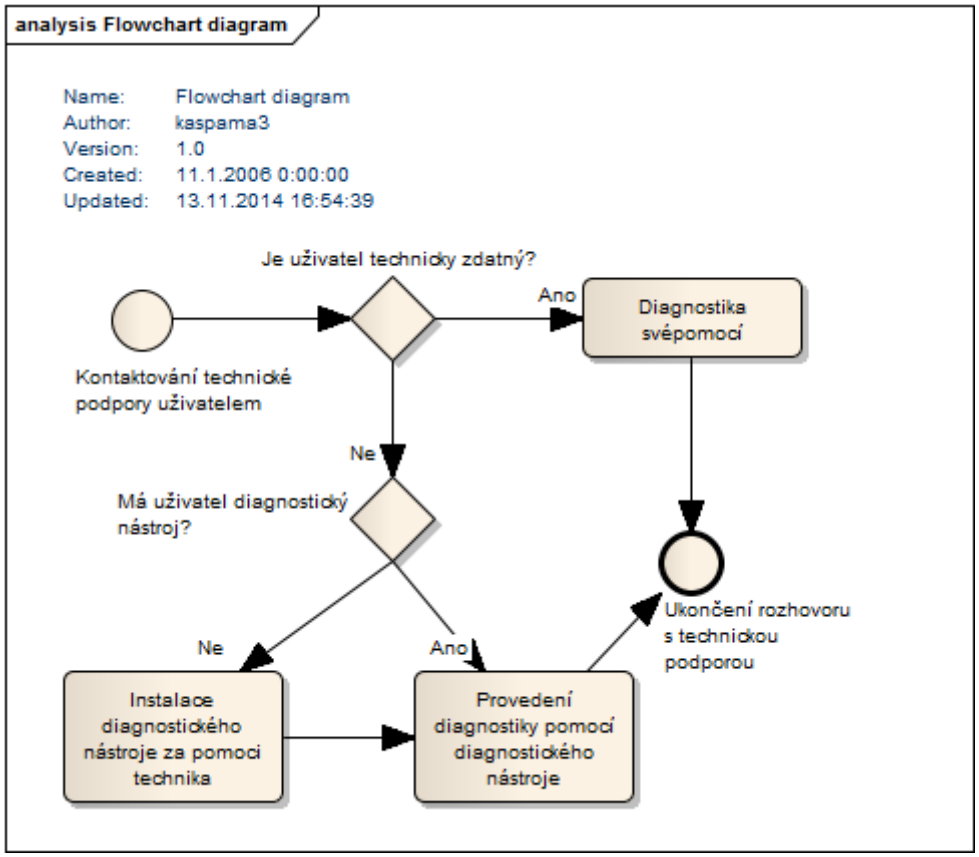
Z předešlého průzkumu používaných systémů vyplývá, že důraz na monitoring sítě je kladen ve směru správců ke koncovým uživatelům. Tento jev je přisuzován hlavně historickému vývoji spolku, kdy zakládající členy tvořili ze sta procent počítačoví nadšenci a lidé zapálení a ochotní porozumět všem novým technologiím. Ti postupně vybudovali a dodnes spravují páteřní infrastrukturu sítě. Ta v současné době dosahuje obřích rozměrů (na komunitní projekt) a není schopna bez dostatečného dohledu řádně fungovat. Toto mají na svědomí dva navzájem se podporující faktory. Na jedné straně technologický boom a značné investice do nových technologií, na druhé straně tímto podporovaný přísun nových členů spolku, kteří přinášejí nové zdroje pro již zmíněné investice. Zákonitě se tak členy stávají i méně počítačově zdatní jedinci a dochází k rozvrstvení společnosti na správce sítě, aktivní členy a koncové uživatele, kterých je nejvíce a mnohdy nemají o principech fungování sítě potřebné znalosti. Na pomoc s jejich připojením do sítě a samotnému fungování v síti byla zřízena technická podpora (jak již bylo zmíněno při představování HKfree), která s nimi denně řeší nejrůznější problémy v otázkách funkčnosti připojení.

Zde bylo identifikováno slabé, a s růstem připojených členů, se stále prohlubující místo v zaběhlém systému fungování. Pracovníci technické podpory evidují denně desítky příchozích telefonických hovorů, jejich náplní je nepřiměřeně zdlouhavá diagnostika stavu připojení ze strany koncového uživatele, vzniklá popisováním základních postupů jako jsou například, měření rychlosti propustnosti sítě, odezvy na hlavní body sítě, správné nastavení síťových adaptérů a mnoho dalších. Z rozhovoru s pracovníky technické podpory bylo jasně patrné, že jakékoliv zlepšení v tomto směru by mělo za následek jejich časové uvolnění v řádech až hodin denně. Tento čas by mohl být efektivně využit na inovace/opravy sítě a pomoc s připojením nových zájemců.

5.4 Prostor pro inovace

Z předchozího chování běžných koncových uživatel je patrné, že textové návody umístěné na veřejně přístupném portálu HKfree, nemají pro řešení situace žádný význam. V případě správného fungování síťového připojení jim nikdo nepřikládá žádnou důležitost, a když něco přestane fungovat, jsou již většinou nedostupné, nebo si situace žádá rychlejší řešení. Za předpokladu, že jen velmi málo jedinců je ochotno prohloubit své znalosti tímto směrem, což je způsobeno klesajícími vstupními nároky na nové uživatele, bylo identifikováno jako nejschůdnější řešení přesunutí těžiště diagnostiky na jejich domácí počítače. Bude tak zamezeno nedostupnosti stěžejních informací a následně ulehčeno pracovníkům technické podpory. Ti by měli již při první návštěvě v místě bydliště zajistit přítomnost vzniklého problému na zařízeních zprostředkovávajících interakci se sítí HKfree, včetně prvotního zaškolení obsluhy a celkového seznámení s danou situací.

Při samotném kontaktování podpory za účelem řešení problémů se stávajícím síťovým připojením, se již bude řídit rozhovor nebo emailová komunikace, striktně dle nastavených procesů, aby bylo dodrženo cílové snížení časové náročnosti. Uživatel se obrátí na podporu s prosbou o pomoc z důvodu nesprávné funkčnosti do sítě a internetu, pracovníkem bude ověřena technická zdatnost kontaktující osoby na základě několika věcných otázek a vyhodnocen způsob dalšího postupu. V případě kladného vyhodnocení situace bude událost řešena na základě přímé komunikace mezi technikem a uživatelem. V opačném případě bude zjišťována přítomnost diagnostického aparátu a podniknou se další příslušné kroky. Pokud bude nedostupný, bude provedena řízená instalace na základě pokynů nebo vzdálené správy, jinak bude rovnou započat proces diagnostiky dle pokynů pracovníka. Na obrázku 11 je graficky znázorněn navrhnutý proces pomocí vývojového diagramu.



Obrázek 11 Schéma provádění diagnostiky
Zdroj: Autor

Z uživatelského pohledu bude tímto zajištěn hladký průběh vyřízení požadovaných informací technického charakteru ze strany pracovníků technické podpory a podpořen tak bezkonfliktní a rychlý působ komunikace obecně.

Aby bylo navrhované řešení komplexnější a ještě více zhodnotilo svůj vývoj, může pomoci již stávajícím monitorovacím systémům. Diagnostický nástroj shromáždí na počítači koncového uživatele důležité informace o koncovém bodu, a ty zanese do své serverové části. Tyto údaje bude nadále možné automatizovaně zpracovat jako vstupní parametry ostatních systémů. Jako nejvýhodnější spolupráce se jeví sdílení informací, a to konkrétně IP adres koncových uživatelů, se systémem SmokePing. Díky tomuto mechanismu zde bude možné spravovat hlídání dostupnosti a latencí koncových bodů. Způsoby distribuce vstupních parametrů do ostatních systémů z jednotného úložiště, bude nutno analyzovat, a vyvinout jako vylepšení monitorovacího aparátu a nebude součástí této práce.

6 Návrh řešení

Jako nejefektivnější možné řešení pro výrobu klientského diagnostického aparátu bylo zvoleno naprogramování samostatné aplikace pro koncové stanice, spolupracující se serverovou podpůrnou částí a v případě výpadků sítě, využívající okolní klientské aplikace jako zdroj informací.

Klientská aplikace – bude program navržený pro instalaci na koncový počítač, zprostředkovávající diagnostické operace směrem od koncového uživatele zpět do sítě, zobrazující aktuální zprávy o dění v síti, a zahrnující jiné podpůrné funkcionality

Serverová podpůrná část – bude obsahovat databázi potřebných hodnot, následně zprostředkovávat komunikaci mezi touto databází a klientskými aplikacemi

Komunikace klient-server – vztah mezi klientskou a serverovou částí řešení, kde dochází k obousměrnému přenosu dat

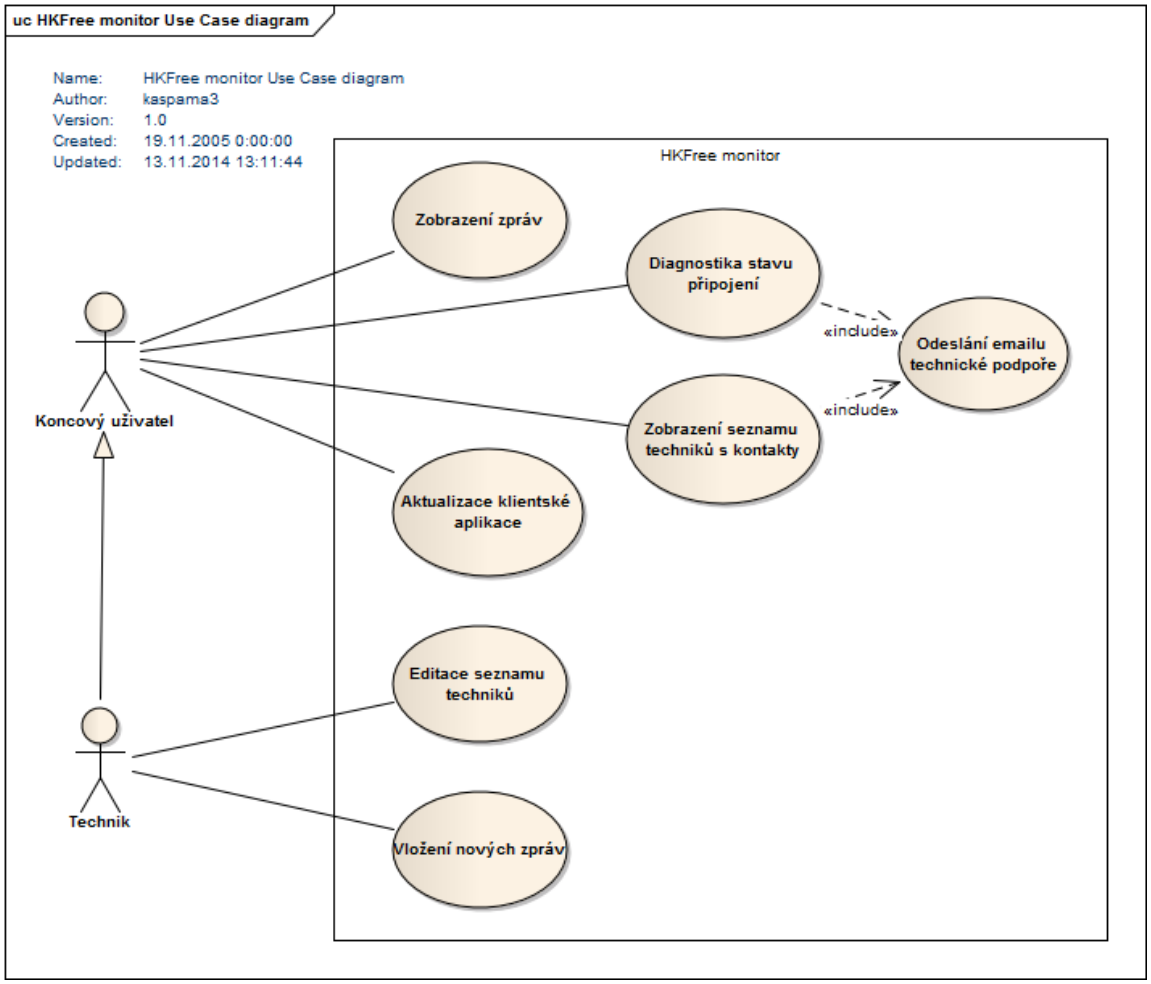
Komunikace klient-klient – vztah mezi dvěma klientskými aplikacemi, kde dochází k obousměrnému přenosu dat

6.1 Případy užití

Na obrázku 12 jsou znázorněny případy užití (Use Case) diagnostického nástroje HKfree monitor, které představují všechny plánované funkčnosti do první verze aplikace. Po delší době používání v praxi bude provedena analýza požadavků, zpracován návrh na úpravy a implementaci nových potřebných funkcí. Toto již nebude předmětem diplomové práce, ale budoucího vývoje aplikace za spolupráce s HKfree a jejich technickou podporou.

Systém je navrhnut pro používání dvěma typy uživatelů, jimiž jsou technik a koncový uživatel. Koncoví uživatelé budou mít ze svých desktopových aplikací

přístupné běžné funkce pro diagnostiku a kontaktování technické podpory, zatímco technici budou mít k těmto funkcím navíc ještě přístupné ovládání uložených informací v serverové části.



Obrázek 12 Use Case diagram monitorovacího aparátu
Zdroj: Autor

6.2 Použité technologie

Následující technologie byly použity při zpracování návrhu a samotném vytvoření diagnostického aparátu.

GNU/Linux

Linux je licenčně volně šiřitelný operační systém postavený na principech Unixu. Díky své jednoduchosti, síle a dostupnosti veškerého programového vybavení, které lze upravovat dle libosti, je mnohdy jasnou volbou pro serverové stroje.

MySQL

Jednoduchý, volně šiřitelný a multiplatformní databázový systém společnosti Oracle. Komunikace s ním probíhá pomocí SQL jazyka, důraz je kladen hlavně na rychlost.

Apache

Jedná se o nejrozšířenější softwarový webový server, který zprostředkovává internetovým prohlížečům požadované internetové stránky.

PHP

PHP je skriptovací programovací jazyk, používaný především k programování dynamických internetových stránek a webových aplikací. Při použití PHP pro tvorbu dynamických stránek se provádějí scripty na straně serveru a k uživateli je přenesen pouze jejich výsledek.

C#

Objektově orientovaný programovací jazyk vyvinutý firmou Microsoft, který se používá pro tvorbu počítačových aplikací, webových aplikací, databázových aplikací atp.

.NET Framework

Běhové prostředí pro aplikace psané pro .NET, obsahující celou řadu podpůrných objektově orientovaných tříd usnadňujících mnoho úkolů.

XML

Je obecný značkovací jazyk, určený především pro výměnu dat mezi aplikacemi a publikováním dokumentů, u kterých popisuje strukturu z hlediska věcného obsahu.

6.3 Návrh klientské aplikace

Klientská část diagnostické a monitorovací aplikace bude instalována běžným způsobem na koncové zařízení uživatele (počítač s OS Windows) a za standartních podmínek nastavení, bude automaticky spouštěna při přihlášení uživatele do systému tak, aby co nejlépe pokryla jeho aktivní dobu strávenou na počítači a zajistila tak informovanost o plánovaném a aktuálním stavu sítě. Ten bude navenek prezentovat formou barevně odlišených ikon zobrazených na panelu nástrojů nabídky Start a v reálném čase reagujících na změny stavu v síti nebo dostupnosti nových rozesílaných zpráv. Z tohoto místa bude taky dostupná kontextová nabídka celé aplikace, která bude tematicky rozčleněna do příslušných nabídek, zobrazovaných v samostatném aplikačním oknu.

6.3.1 Diagnostika

Bezesporu nejzásadnější částí aplikace je diagnostická funkce a celková pomoc uživateli zhodnotit stávající stav sítě, který zobrazí jako výstup. Pomocí tohoto nástroje budou sledovány následující hlavní parametry připojení:

- dostupnost internetu
- dostupnost hlavního počítače HKfree
- dostupnost první brány v HKfree
- dostupnost brány počítače

Na základě těchto zjištěných hodnot již bude uživatel snadno schopen reportovat stav svého připojení a pomoci tak k zdárnému vyřešení situace. Stav parametrů dostupnosti bude ověřován za pomoci příkazů Ping a Tracert, představených v teoretické části práce.

Dostupnost brány počítače bude vyhodnocena na základě odpovědi Ping zaslané na její síťovou adresu zjištěnou programově, pomocí zabudovaných knihoven prostředí .NET a určené logiky. Tato zásadní informace napoví, zda-li je počítač připojen korektně ke svému prvnímu směrovači. Ve většině případů je tento směrovač představován domácím přípojným bodem, běžně nazývaným „AP“ (z angl. access point) nebo „wifi router“ (většinou poskytuje pokrytí bezdrátovým Wi-Fi signálem) a může odhalit jeho nefunkčnost, samovolné resetování nastavení, absenci bezdrátového signálu, vypnutí adaptéru bezdrátového připojení, poškození síťového kabelu atp. Ve velmi málo případech je počítač napojen přímo do sítě HKfree bez předsazeného „domácího routeru“. Adresa brány počítače se bude shodovat s adresou první brány v síti HKfree a bude možno předchozí domněnky eliminovat.

Dostupnost první brány v HKfree bude vyhodnocena na základě odpovědi Ping na její síťovou adresu zjištěnou dle následujícího postupu. Celá síť využívá jednotný adresní prostor s prefixem „10.107“, porovná-li se výstup Tracertu s tímto prefixem, tak první výskyt adresy splňující tuto podmínku je požadovaná síťová adresa. Dle její dostupnosti, lze uvažovat dostupnost připojení do sítě HKfree. Většina domácností je připojena pomocí bezdrátové technologie WiFi na přípojné body HKfree. Zde mohou být identifikovány potíže jako jsou rušené vysílací kanály, blokování připojení na přípojný bod, otočená anténa vysílače z důvodu povětrnostních vlivů atp. Menší, ale nezanedbatelné množství domácností, je připojeno na první směrovač HKfree ethernetovým kabelem, a to z důvodu napojení na LAN síť ve velkých panelových domech. Zde může představovat potíže fyzicky přerušená kabeláž.

Dostupnost hlavního počítače HKfree bude vyhodnocena na základě odpovědi Ping na jeho síťovou adresu, která je pevně stanovena. Pokud bude potvrzena prostupnost až do tohoto bodu, tak všechny síťové prvky na trase až k hlavnímu počítači jsou v pořádku. V opačném případě je nutno pátrat napříč páteřní infrastrukturou, a k tomu bude nápomocen celkový výpis Tracert, zobrazující síťové adresy až do posledního funkčního bodu.

Dostupnost internetu bude vyhodnocena na základě odpovědi Ping na adresu www.google.com, vnímanou jako jednu z veřejně využívaných ke sledování dostupnosti. Členství v síti má dvě podoby, a to přístup pouze do vnitřní sítě a na stroje v ní umístěné, nebo s přístupem do Internetu. Jedná-li se o běžného člena s povoleným přístupem do Internetu, hlavním důvodem nedostupnosti bývá nezaplacení členských příspěvků a následné zablokování automatickým systémem. V menší míře mohou být na vině plánované či neplánované odstávky nebo v horším případě poruchy.

Předešlé 4 typy dostupnosti budou v aplikačním okně po vyhodnocení označeny jako splněné či nesplněné a doplněny výpisem testu dostupnosti Ping pro diagnostiku spolehlivosti a odezvy.

```
Pinging 10.107.4.100 with 32 bytes of data:

Reply from 10.107.4.100: bytes=32 time=2ms TTL=61
Reply from 10.107.4.100: bytes=32 time=1ms TTL=61
Reply from 10.107.4.100: bytes=32 time=2ms TTL=61
Reply from 10.107.4.100: bytes=32 time=2ms TTL=61

Ping statistics for 10.107.4.100:
        Packets: Sent = 4, Received = 4, Lost = 0
Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 2ms
```

Obrázek 13 Výstup Ping diagnostické části aplikace
Zdroj: Autor

Doplňkovým diagnostickým testem je test rychlosti přenosu dat, který se automaticky ve formě přenosu o objemu dat 1MB zařadí při dostupnosti hlavního počítače HKfree mezi ostatní prováděné testy. Po provedení zobrazí dobu potřebnou k přenesení souboru z hlavního počítače do klientského spolu s vyčíslenou přenosovou rychlostí. Test rychlosti přenosu dat o objemu 10MB bude možno spustit ručně z důvodu větší časové náročnosti hlavně u pomalejších síťových připojení.

Po úspěšném vykonání testu diagnostiky, bude možno zobrazit uložené běžné postupy při odstraňování závad, nasbírané praxí pracovníků technické podpory. Dle nich si bude moci uživatel sám zkontrolovat běžně dostupné věci, které by mohl případně napravit. Pokud nebude problém v síti možné vyřešit svépomocí, sdělí pracovníkům technické podpory naměřené hodnoty, které budou moci analyzovat a navrhnout řešení. V případě dostupnosti hlavního počítače HKfree a tím pádem i SMTP serveru pro odesílání emailové pošty, bude dostupná funkce odeslání všech diagnostických výstupů v jedné emailové zprávě na adresu technické podpory.

6.3.2 Zprávy

Funkcionalita modulu zpráv bude sloužit k rozesílání informací o aktuálním či plánovaném budoucím stavu sítě. Její zapracování nahrazuje rozesílání zpráv emailem nebo vystavování na diskusní skupiny, jež jsou v době největší potřeby zjištění aktuálního stavu v důsledku nefunkčnosti sítě mnohdy již nedostupné. Mechanismus byl navržen tak, aby se sám přizpůsobil vzniklé situaci a pokud možno, dopravil informace až na koncový počítač.

Ve většině případů bude aplikace kontrolovat v přednastaveném intervalu dostupnost nových zpráv na serveru na základě dostupnosti hlavního počítače a fungování aplikace v režimu plné dostupnosti. Jakmile administrátor uloží na server novou zprávu a kontrolující cyklus ji vyhodnotí ke stažení, bude uložena do zvláštního konfiguračního souboru klientské aplikace a publikována jako nová a nezobrazená. Ikona na panelu nástrojů bude signalizovat novou nepřečtenou zprávu až do jejího zobrazení a nad ní se zobrazí na několik vteřin tooltip nabídka (bublina s popisem zobrazená nad ikonou) s obsahem.

Při nedostupnosti serverové části se přepne zabudovaný mechanismus kontroly zpráv do režimu klient-klient (peer-to-peer) komunikace, a bude se snažit získat zprávy od okolních klientských aplikací. Tento postup zajistí aktuálnost informací i počítačům, které byly například vypnuté a neměly možnost si stáhnout informace o plánovaných výpadcích ze serveru ještě před tím, než výpadek nastal. Pro takový

způsob komunikace je překážkou hojně používaný NAT v síti (představený v teoretické části práce), který prakticky znemožňuje přímou komunikaci mezi dvěma takto „schovanými“ zařízeními. Více podrobně bude tato problematika rozebrána v samostatné části.

6.3.3 Podpora

Modul Podpora bude tvořit část programu, kde budou na jednom místě shrnuty následující kontakty technické podpory:

- jméno a příjmení
- telefon
- email
- ostatní (Facebook, ICQ, jabber…)

Při každém spuštění programu bude v případě dostupnosti vždy kontrolován aktuální seznam na serveru a stahován do klientské aplikace tak, aby byla zajištěna jeho stoprocentní spolehlivost pravosti informací.

Komfort a pohodlí při kontaktování podpory zde zajistí i přítomnost formuláře pro odeslání emailu s předvyplněnou částí hlavičky pro snazší identifikaci uživatele a jednotnost formátu zpráv kvůli filtrování v poštovním klientu, spolu s rozbalovacím menu emailových adres techniků pro zaslání bez nutnosti vypisování adres ručně.

6.3.4 Aktualizace

Z důvodu postupného zdokonalování a inovování bude zapracován mechanismus kontroly dostupnosti nových verzí na předem určeném úložišti. Stiskem jediného tlačítka bude možné aplikaci zaktualizovat, nebo po zapnutí automatických aktualizací bude proveden tento proces sám od sebe.

6.3.5 Nastavení

Uživatelsky nejméně využívaná část aplikace zajišťující nastavení intervalů pro kontrolu nových zpráv, dostupnosti automatických updatů a stavu aplikace. Dále potom záznam o přidělené IP adrese v rámci sítě HKfree a funkci mazání konfiguračních souborů z důvodu celkového resetu a vygenerování nových konfiguračních souborů.

6.4 Návrh serverové části

Serverová část řešení bude umístěna v centrálním bodě spolku HKfree, tak aby byla zajištěna co nejlepší dostupnost ze všech částí sítě. Důraz je kladen hlavně na funkčnost a jednoduchost obsluhy, kterou využijí výhradně pracovníci technické podpory a jejich občasné zastoupení. Plánovanou funkčností bude napojení na současný LDAP server a použito ověřování na základě již používaných přístupů sdílených pro většinu zaběhlých systémů. Dále budou zajištěny 3 základní funkce pro obsluhu vzniklého řešení za pomoci webových stránek administračního rozhraní:

- obsluha rozesílaných zpráv
- obsluha uložených kontaktů
- obsluha distribuce aktualizací

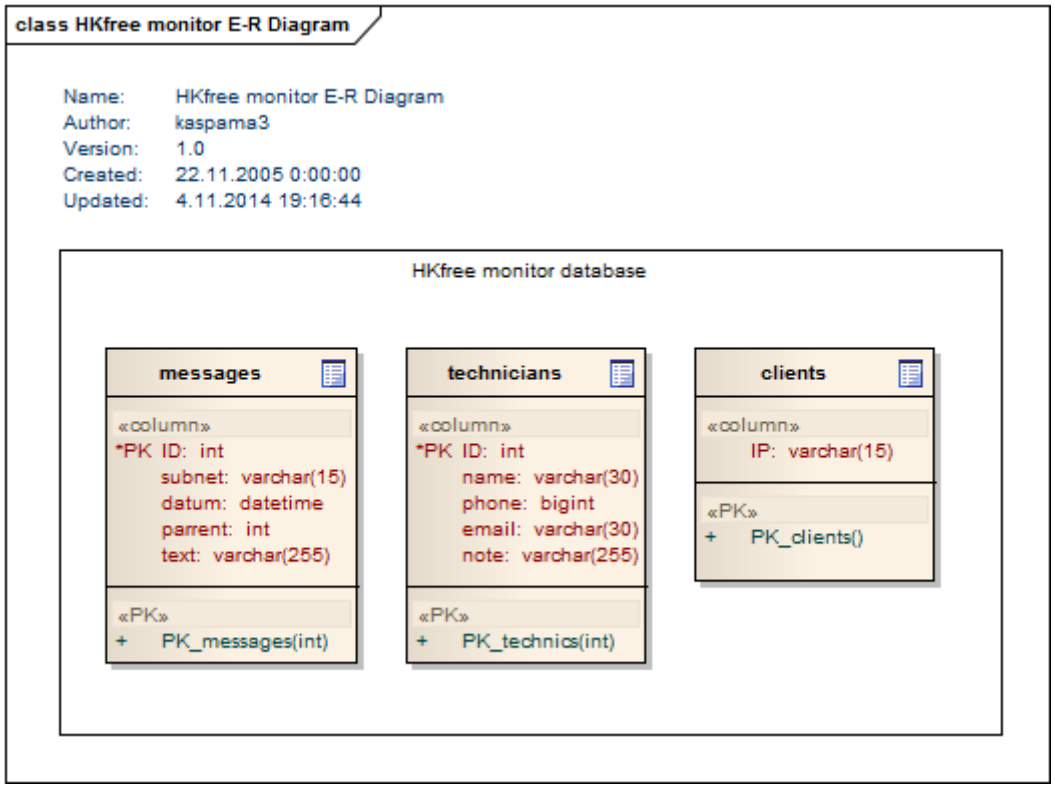
Pro ukládání dat bude použita MySQL databáze na společném databázovém serveru za účelem centralizace a jednodušší správy zdrojů.

6.5 E-R diagram databáze

Pro účely vývoje aplikace byl sestaven E-R diagram databáze. Ta bude zpočátku obsahovat 3 tabulky:

1. **Messages** – zprávy zasílané do klientských aplikací
2. **Technicians** – údaje o stávajících pracovnících linky technické podpory
3. **Clients** – adresy koncových počítačů/klientů

Nejsou zde naznačeny žádné vazby mezi jednotlivými tabulkami, protože databáze je pro první verzi aplikace navržena velice jednoduše a tabulky na sobě nejsou závislé.



Obrázek 14 E-R Diagram databáze
Zdroj: Autor

6.6 Způsob komunikace

Komunikační procesy probíhající v diagnostickém aparátu lze rozdělit na dva způsoby:

- klient-server
- klient-klient

Komunikace klient-server reprezentuje předávání informací mezi klientskou aplikací a serverovou podpůrnou částí. Přenášená data jsou nejednotného charakteru. Na server jsou odesílány buďto samotné požadavky, nebo požadavky

s přiloženou specifikací, a zpět ze serveru jsou odesílány textové řetězce nebo celé tabulky. Z tohoto důvodu byla zvolena forma přenosu pomocí XML a definována přenosová struktura. Iniciace odpovědi ze serveru byla nejdříve testována pomocí funkce PHP GET, ale jako limitující se ukázala maximální možná délka přenášených dat 512 znaků. Proto byl dále vývoj směrován k použití PHP POST metody, která nabízí neomezený počet znaků v přenosu dat a lepší podmínky šifrování přenosu obecně. Celý postup výměny informací začíná inicializací ze strany klientské části pomocí PHP POST metody, která je serverem zpracována a odpověď je přenesena formou XML. Jako příklad je uvedena struktura pro stahování zpráv:

```
<?xml version="1.0" encoding="WINDOWS-1250"?>
<LIST version="1">
<ITEM ID="1" SUBNET="all" DATE="2014-06-12" PARRENT="0" TEXT="1. zpráva" />
</LIST>
```

XML dokument je uveden obecnou hlavičkou a obsahuje element „LIST“, který reprezentuje seznam přenášených hodnot a jeho atribut „VERSION“, v tomto konkrétním případě značí nejnovější dostupnou zprávu na serveru. List obsahuje elementy „ITEM“, reprezentující položky přenášeného seznamu a mající určité identifikační atributy.

Komunikace klient-klient (peer-to-peer) představuje proces výměny informací mezi jednotlivými klientskými aplikacemi v případě nedostupnosti serverové části, která je vyhodnocena z aktuálního pracovního stavu aplikace

Za tímto účelem byl nejdříve testován UDP přenos dat, ale nevyhověl svou povahou požadovaným parametrům. Nejzávažnějším problémem je nepřítomnost zpětné vazby komunikačního protokolu, a tím pádem absence jakékoliv informace o stavu kontaktovaného zařízení. Další vývoj byl veden pomocí TCP spojení. Každá klientská aplikace je zároveň server naslouchající na určitém portu příchozí spojení a zároveň klient posílající žádost o spojení. Po úspěšném vytvoření tunelu

dojde k obousměrné komunikaci, výměně dat a uzavření tunelu. Tunel nezůstává otevřený z důvodu komunikace mezi více klienty a nastavitelným intervalem samotné kontroly nových informací. V opačném případě by bylo generováno spoustu nevyužitých spojení v síti, a zbytečně by byly blokovány prostředky, nehledě na to, že různé síťové prvky by mohly vyhodnotit situaci jako útok a blokovat. Technicky je komunikace TCP zajištěna tak, aby jeden server mohl aktivně komunikovat s více klienty najednou.

Samotné komunikaci mezi klienty předchází jejich vzájemnému vyhledání. V prvotní fázi byly testovány různé způsoby prohledávání sítě a lokalizace aktivních koncových aplikací. Tento způsob byl poněkud nešťastný v tom, že se snažil kontaktovat veliké množství adres a pokročilejšími přepínači v síti byl tento provoz vyhodnocen jako škodlivý a tudíž blokován na linkové vrstvě. Nyní koncová aplikace již při vytváření konfiguračního souboru odešle na server informaci o svojí adrese, na serveru jsou uloženy adresy všech těchto aplikací a ty jsou následně stahovány zpět do aplikací jako seznam klientů pro kontaktování. Tyto seznamy jsou udržovány stále aktuální. V tuto chvíli je model nastaven tak, že zjišťuje informace od všech klientů v síti, v případě nárůstu jejich počtu pak bude nutné redukovat jejich seznam. Jedna z možností je porovnávat seznam s výpisem funkce tracert – všechny podsítě cestou k hlavnímu počítači a dotazovat se pouze těchto zařízení s předpokladem, že jako sousední aplikace budou vědět o problematice dané lokality. Další možnost by bylo implementovat mapu sítě z OSPF mapy (představené v kapitole „Síťová infrastruktura“) a dotazovat se klientů společné celé větve podsítě. Toto již ale není součástí této práce a bude zahrnuto jako jeden z plánovaných směrů rozvoje diagnostické aplikace.

Jak již bylo představeno v návrhu aplikace, problém komunikaci představuje NAT, z jehož definice vyplývá, že pokud požadavek na spojení nepřichází směrem z vnitřní sítě, nebude možné spojení navázat, protože nelze napřímo adresovat cíl schovaný v natované síti. Díky této komplikaci bude zajištěna stoprocentní spolehlivost distribuce zpráv v odříznuté části sítě pouze za předpokladu, že bude existovat alespoň jedna klientská aplikace, která si ještě před výpadkem stáhla

aktuální zprávy ze serveru a není součástí „zanatované“ privátní sítě tak, aby jí ostatní klientské aplikace v okolí mohly kontaktovat přímo, a získaly od ní aktuální zprávy o stavu sítě a plánovaných výpadcích. Takto získané informace budou zpracovány stejným způsobem, jako při stažení ze serveru.

V teoretické části byl představen i nástroj, s jehož pomocí je možné NAT obejít a vytvořit tak spojení mezi dvěma koncovými počítači. Toto řešení však nebylo do práce zahrnuto z důvodu rozsáhlejší složitosti problému a bude zahrnuto jako jeden z plánovaných směrů rozvoje diagnostické aplikace.

6.7 Aktualizace

V konfigurační části klientské aplikace bude zapracován záznam o verzi programu. Při vyvolání metody pro kontrolu dostupných aktualizací dojde ke zjištění nejnovější dostupné verze na internetovém úložišti, která bude porovnána s aktuální instalovanou. Pokud se verze budou lišit, spustí se aktualizační proces. Metodu kontroly dostupných aktualizací bude možno vyvolat ručně nebo pomocí funkce automatických aktualizací, obsluhované časovačem a spouštěné v pravidelných intervalech.

Aktualizační proces nejdříve prohlédne složku s updaty v instalačním adresáři klientské aplikace a odstraní případné staré aktualizace a pomocné nástroje. Dále z internetového úložiště stáhne nový aktualizační balíček a pomocný nástroj, který spustí. Pomocný nástroj ihned po spuštění lokalizuje a ukončí proces klientské aplikace, následně rozbalí aktualizační balíček a přepíše staré soubory novými. Jakmile dokončí vše potřebné, spustí opět klientskou aplikaci a sám se ukončí.

7 Implementace

Celé programování aplikace nebude v této práci popisováno, budou vybrány a okomentovány pouze zajímavé zdrojové kódy. Kompletní zdrojová data budou dostupná na přiloženém médiu.

7.1 Popis zajímavých částí programu

7.1.1 Aktualizace seznamu klientských adres

Klientská aplikace při každém spuštění zkontroluje aktuálnost svého seznamu klientských aplikací pro vzájemnou komunikaci. Děje se tomu tak voláním metody „checkClients“, která si vytvoří prázdný list klientských adres a následně do něj uloží výsledek komunikace se serverem, zprostředkovaný pomocí metody PHP Post. Jako vstupní parametry je použita uložená internetová adresa obslužného PHP scriptu, proměnná Post požadavku a aktuální checksum (kontrolní součet) uložených adres. Porovnání seznamu uloženého na serveru a seznamu v aplikaci probíhá pomocí porovnání jejich spočítaného checksum. Pokud server vrátí neprázdný seznam adres, aplikace s ním nahradí svůj seznam, uloží jeho nový checksum a uloží změněnou konfiguraci do konfiguračního souboru.

```
internal void checkClients()
        {
            List<string> tmpClients = new List<string>();
            var cli =
xc.clientsXMLparser(WebComm.fastPHPpost(conf.phpAddr, "clients",
conf.clientsChecksum.ToString()));
            if (cli.Count() > 0)
            {
                conf.clientsChecksum = xc.clchecksum;
                conf.clients = cli;
                conf.saveConfiguration();
            }
        }
```

Serverová část zpracovává dotaz skriptem na základě zaslané IF podmínky, v tomto případě tedy požadavek „clients“. Nejprve dojde k výpočtu checksum hodnoty tabulky obsahující seznam klientských IP adres a jeho porovnání s se zaslaným checksum z klientské aplikace. Pokud je podmínkou IF identifikován rozdíl, dojde k vygenerování hodnot z databáze do XML struktury pro přenos dat

mezi aplikací a serverem a její odeslání zpět do aplikace. V opačném případě je vráceno „empty“, značící aktuálnost dat.

```
if( $_POST["clients"] ) {
    $serverChecksum = "empty";
    $sql = "CHECKSUM TABLE clients";
    $result   = mysql_query($sql) or die(mysql_error());
    while ($row = mysql_fetch_assoc($result)) {
        $serverChecksum = $row['Checksum'];
    }
    $clientChecksum = $_POST['clients'];
    if ( $clientChecksum <> $serverChecksum) {
        $sql2 = "SELECT * FROM clients";
        $q2   = mysql_query($sql2) or die(mysql_error());
        $xml2 = "<?xml version=\"1.0\"?>\n";
        $xml2 .= "<LIST version=\"".$serverChecksum."\">\n";
        while($r2 = mysql_fetch_array($q2)){
            $spIP = htmlspecialchars($r2['ip']);
            $xml2 .= "<ITEM IP=\"".$spIP."\" />\n";
        }
        $xml2 .= "</LIST>";

        echo $xml2;
    }
    else {
        echo "empty";
    }
}
```

Poslední součástí tohoto procesu je transformace přijatého XML do třídy reprezentující seznam klientských IP adres pro její následné uložení. Tento parser se opakuje ve více částech programu s vlastnostmi upravenými speciálně pro danou komunikaci a jeho obdoba bude představena v dalších částech kódu.

V současné době umí aplikace kontrolovat seznam klientských IP adres pouze proti serverové části. V budoucnu je tato možnost plánována i pro komunikaci mezi klientskými aplikacemi a bude zapracována jako jedna z aktualizací. Implementace této funkčnosti nebude náročná, půjde o pouze o jiné použití stávajících mechanismů a jejich propojení.

7.1.2 Kontrola dostupných zpráv

Dostupnost nových zpráv je kontrolována v pravidelných časových intervalech pomocí zabudovaného časovače (Timer), jehož interval cyklu je nastavitelný

v aplikačním modulu „Nastavení“. Po zapracování této funkčnosti a jejím následným testováním bylo zjištěno, že aplikace se po dobu kontaktování nedostupných klientských aplikací a čekání na odpověď, stává nedostupnou. Tato doba je přímo úměrná četnosti adres nedostupných klientských aplikací a intervalu kontroly zpráv. Tato vlastnost byla odstraněna spouštěním metody obsluhující kontrolu zpráv („loadMessages“) v samostatném vlákně.

```
Thread t = new Thread(new ThreadStart(checkMessages));
            t.IsBackground = true;
            t.Start();
```

Samotná metoda potom založí seznam, jehož prvky tvoří instance třídy „MessagStruct“, která definuje strukturu jednotlivých zpráv pro další zpracování, a na základě podmínky IF vyhodnotí, v jakém režimu má kontrolovat nové zprávy. Pokud je „appState“ roven nule, znamená to, že aplikace je ve stavu, kdy nemá možnost komunikovat po síti HKfree a snaží se kontaktovat okolní klientské aplikace. Pokud je „appState“ různý od nuly, pracuje aplikace v režimu dostupnosti Internetu nebo alespoň hlavního počítače HKfree, a kontaktuje pro stažení nových zpráv serverovou část. V kódu je možné si povšimnout zapracovaného pravidla „File.Exists“, které v případě dostupnosti souboru s názvem „Debug“ přepíná aplikaci do režimu komunikace peer-to-peer pro testovací účely.

```
internal void checkMessages()
        {
            List<messageStruct> zpr;
            if (appState == 0 || File.Exists(conf.appStart +
"\\Debug"))
            {
                foreach (string cl in conf.clients)
                {
                    zpr =
xc.messagesXMLparser(pc.getListSync("$message;" + msg.getLatest(),
cl));
                    addMessages(zpr);
                }
            }
            else
            {
                zpr =
xc.messagesXMLparser(WebComm.fastPHPpost(conf.phpAddr, "messages",
msg.getLatest().ToString()));
                addMessages(zpr);
            }}
```

Kontrola zpráv proti serverové části probíhá velice podobně jako kontrola klientských adres. V případě komunikace mezi klientskými aplikacemi peer-to-peer dochází k vytvoření TCP spojení, jehož zajišťující zdrojový kód je k nahlédnutí v Příloze č. 1 a odeslání požadavku pro příjem nových zprav uvozeného prefixem „$message;“ a doplněného ID poslední známé zprávy. Tato spojení jsou pomocí cyklu navázány postupně pro všechny klientské IP adresy vybrané pro komunikaci.

Zde je prostor pro budoucí rozšíření mechanismu výběru relevantních adres kontaktovaných klientů. Jedna z možností by byla již zmíněné porovnávání seznamu s trasováním síťové cesty k hlavnímu počítači. Složitější řešení by mohlo být ukládání informací o úspěšnosti komunikace a vytvoření žebříčku nejspolehlivějších klientů. Tato zásadnější změna by měla za následek vytvoření třídy představující klienta, která by obsahovala IP adresu a pole pro uložení hodnocení. Dále by pak následovala úprava metod zajišťujících ukládání klientů jak pro udržení těchto hodnot, tak pro samotnou práci s nimi, ve smyslu navyšování nebo ponižování hodnocení a v neposlední řadě také úprava databázové tabulky. Mechanismus výběru klientů pro kontaktování by se pak řídil hodnocením a v první řadě by kontaktoval nejspolehlivější klientské aplikace (ty, které bývají nejčastěji dostupné).

Kontaktovaná klientská aplikace, nyní v pozici serverové části, naslouchá příchozí spojení pomocí spuštěného TCP serveru a přijatá data dále zpracovává. Dle prefixu „$message;“ pozná, že se jedná o požadavek na nové zprávy a následná informace je parsována jako číslo reprezentující ID poslední zprávy. Následně aplikace sestaví XML strukturu a pokud má k dispozici novější zprávy, tak jimi strukturu naplní. V případě neprázdného seznamu je odeslána zpět XML struktura. V opačném případě pouze řetězec „empty“, značící nedostupnost nových zpráv.

```
if (data.StartsWith("$message"))
                {
                    string outputStart = "<?xml version=\"1.0\"
encoding=\"WINDOWS-1250\"?><LIST version=\"3.2.1\">";
                    string outputEnd = "</LIST>";
                    data = data.Substring(data.IndexOf(';') + 1);
                    string list ="";
```

```
                    foreach (messageStruct ms in msg)
                    {
                        if (ms.ID > conf.tryParseInt(data)) list =
list + "<ITEM ID=\"" + ms.ID + "\" SUBNET=\"" + s(ms.subnet) + "\"
DATE=\"" + ms.datum + "\" PARRENT=\"" + s(ms.parrent) + "\" TEXT=\"" +
s(ms.text) + "\" />";
                    }
                    string output;
                    if (list != "")
                    {
                        output = outputStart + list + outputEnd;
                    }
                    else output = "empty";
                    this.writeData(so,
Encoding.GetEncoding(0x4e2).GetBytes(output + ";"));
                    return;
                }
```

V tomto místě je prostor pro další možnou inovaci formou aktualizace programu, a sice prolomení principu NAT pomocí nat-traverse zmíněném v předchozích kapitolách této práce.

Kontrola zakázaných znaků probíhá pomocí následující metody.

```
internal string s(string t)
        {
            return System.Security.SecurityElement.Escape(t);
        }
```

Přijatý seznam zpráv ve formátu XML je následně zpracováván metodou „messagesXMLparser“, která zajistí zpětné rozložení přijatých dat na objekty zprávy „messageStruct“ a vložení do seznamu. Tento postup je zajištěn .NET funkcí pro zpracování XML. Konkrétně se jedná o rozpad dokumentu na definované nody (uzly) a jejich postupné vložení do objektu zprávy.

```
public List<messageStruct> messagesXMLparser(string xmlsoubor)
        {
            List<messageStruct> zpravy = new List<messageStruct>();

            try
            {
                XmlDocument xd = new XmlDocument();
                //xd.Load(xmlsoubor);
                xd.LoadXml(xmlsoubor);
                XmlNodeList xnlrsp = xd.GetElementsByTagName("ITEM");
                foreach (XmlNode xnod in xnlrsp)
                {
```

```
                    zpravy.Add(new
messageStruct(tryParseInt(xnod.Attributes["ID"].Value),
xnod.Attributes["SUBNET"].Value,
xnod.Attributes["DATE"].Value,xnod.Attributes["PARRENT"].Value,
xnod.Attributes["TEXT"].Value));
                }
            }
            catch (Exception)
            {
            }
            return zpravy;
        }
```

Oba způsoby komunikace vracejí seznam přijatých zpráv, který je nadále zpracován metodou, která se nazývá „addMessages“. Ta si deklaruje přepínač „saveMsg“, podle kterého se na závěr provádí uložení zpráv, nastavených na zápornou hodnotu. Když není seznam zpráv prázdný, projde postupně metoda přijaté zprávy a porovná je s uloženými v aplikaci na základě shodnosti ID. Pokud daná zpráva ještě neexistuje, tak ji uloží do seznamu, nastaví přepínač „saveMsg“ na kladnou hodnotu a nad ikonou aplikace zobrazí tooltip zprávu s textem a časem vložení do systému. Tato zpráva je dostupná buď proklikem z tooltip nabídky nebo v modulu Zprávy v menu aplikace. Dokud nebude zpráva zobrazena uživatelem, je ikona programu změněna na obrázek poštovní obálky značící novou nepřečtenou zprávu. Jakmile je cyklus dokončen, tak je nový seznam zpráv uložen do zvláštního konfiguračního souboru na základě stavu přepínače „saveMsg“.

```
internal void addMessages(List<messageStruct> zpr)
        {
            bool saveMsg = false;
            if (zpr == null) zpr = new List<messageStruct>();
            foreach (messageStruct ms in zpr)
            {
                if (!msg.seznamZprav.Any(prod => prod.ID == ms.ID))
                {
                    msg.seznamZprav.Add(ms);
                    saveMsg = true;
                    newMessageAvailable = true;
                    changeIcon();
                    trayIcon.ShowBalloonTip(3000, "Nová zpráva číslo "
+ ms.ID + " ze dne " +
conf.tryParseDate(ms.datum).ToString("dd/M/yyyy"), ms.text,
ToolTipIcon.Info);
                }
            }
            if (saveMsg == true) msg.saveMessages();
        }
```

7.2 Řešené problémy

7.2.1 Instalační adresář

Řešeným problémem je instalační místo aplikace. Nejdříve byla aplikace instalována do složky Program Files, kde se vyskytl problém s konfiguračními soubory. Zabezpečení novějších systémů Windows nedovolí aplikaci ukládat pomocné soubory do tohoto umístění a má vyhrazeno speciální umístění ve složce AppData uvnitř každého profilu. Po předělání ukládání konfigurace tímto způsobem však vyvstaly problémy s aktualizací programu.

Díky zabezpečení systému Windows nelze přepsat stávající soubory aplikace za pomoci spuštění jiné aplikace. Z tohoto důvodu byla celá instalace přesunuta do adresáře AppData po vzoru Google Chrome a vše již probíhá v pořádku.

7.2.2 Aktualizace aplikace

Samotná aktualizace aplikace probíhá za pomoci nástroje updater, který je při vyvolání aktualizace stahován z internetového úložiště spolu s aktualizačním balíčkem. Tento systém je použit, protože aplikace se sama nedokáže přepsat, když je spuštěna. Stáhne tedy updater, který spustí a on zařídí vše potřebné. V následujícím kódu je znázorněno jakým způsobem zastavuje běžící aplikaci.

```
foreach (Process p in
System.Diagnostics.Process.GetProcessesByName("HKFmonitor"))
                {
                    try
                    {
                        p.Kill();
                        p.WaitForExit();
                    }
                    catch
                    {}
                }
                if
(System.Diagnostics.Process.GetProcessesByName("HKFmonitor").Length >
0)
                {
                    throw new Exception("HKFmonitor je spuštěn,
ukončete jej prosím!");}
```

Dále pak dochází k přepsání aktualizovaných souborů jejich náhradami z aktualizačního balíčku. Operace přepsání ale nemusí být z více důvodů úspěšná (např. soubor je stále otevřen atp.), proto se v případě její chyby zkouší zopakovat. Níže je znázorněn zdrojový kód, který tuto funkci obsluhuje včetně opětovného spuštění aplikace.

```
int count = 0;
while (count < 3)
    {
     try
        {
         File.Copy(newfile, updPath, true);
         count = 3;
         var proc = Process.Start(updPath);
        }
     catch (Exception)
        {
         Thread.Sleep(300);
         count++;
        }

    }
```

8 Testování

Při vývoji diagnostického aparátu byly použity následující způsoby testování, které budou nyní představeny spolu s jejich výsledky.

8.1 Virtuální prostředí

Prvotní fáze programování aplikace probíhala samostatně za využití jednoho počítače a nástroje Microsoft Visual Studio 2010, při zpracování peer-to-peer komunikace však muselo být přistoupeno k testování na více počítačích. Důvodem byla potřeba testovat komunikaci mezi více zařízeními přes jejich síťové rozhraní a simulovat tak reálný provoz.

Pro tyto účely byl použit VirtualBox firmy Oracle. Jedná se o bezplatný program pro kompletní simulaci hardwarového prostředí, potřebného pro běh operačního systému. Tímto způsobem byly vytvořeny dva virtuální stroje a do nich nainstalován operační systém Windows 7, reprezentující většinu počítačů používaných v dnešní době. Aplikace neklade žádné nestandartní nároky na hardwarové ani softwarové prostředí a testování nebylo tímto směrem praktikováno.

Pomocí VirtualBoxu byla na hostujícím počítači vytvořena virtuální síť, jejíž účastníci byli mezi sebou propojeni za pomoci jednoduchého bridge a k adresaci síťových karet byl použit adresní prostor 10.107.x.x, představující síť HKfree. Toto je zároveň nutná podmínka pro korektní diagnostickou funkčnost aplikace, z důvodu porovnávání výpisu tras a nastavení počítače. Na komunikaci mezi klienty to nemá žádný vliv.

Samotné testování pak probíhalo formou provozu nainstalované aplikace a analýzou zpráv po pádu nebo hlášení v debuggeru Visual Studia na hostujícím počítači. Mezi hlavní řešené problémy patřily nefunkčnost serverové části a neotevření naslouchacího portu diagnostikovaná neúspěšným telnet připojením na

daný port, blokace celé komunikace firewallem systému Windows nebo chybující identifikace a výběr protějších stran pro sestavení komunikace.

Po doladění chyb a docílení bezproblémové funkčnosti ve virtuálním prostředí byla aplikace postoupena do další fáze testování.

8.2 Reálné prostředí

Pro testování v reálném prostředí byly použity 3 počítače umístěné v různých částech sítě a serverová část umístěná v centrálním bodě HKfree. Parametry použitých zařízení byly stejné jako ve virtuálním prostředí obohacené o síťové prvky tvořící páteř sítě a různorodé technologie přenosu dat.

Zde se projevily potíže způsobené více adresními prostory, které měly za následek chybné diagnostikování dostupnosti některých stěžejních bodů. Všechny nedostatky byly odstraněny na základě zkušebního provozu, centrálně řízeného pomocí připojení ke vzdálené ploše a vzájemné komunikace koncových počítačů. Spolupráce PHP scriptů serverové části a klientských aplikací nepředstavovala problém ani v jednom ze zkušebních testů.

Přestože nebyl zřejmý důvod pro obavy z problémů způsobených objemem přenášených dat, nemohla být tato možnost ani vyloučena. Po sestavení testovacích dat pro přenos mezi klienty i mezi klientem a serverem se však díky principu fungování PHP Post a TCP spojení neprojevily sebemenší známky selhávání.

8.3 Linka technické podpory

Po úspěšném testování a odladění byla aplikace postoupena do rukou pracovníků technické podpory a pěti vybraných koncových uživatelů. Cílem testování bylo zajistit zpětnou vazbu od lidí, pro něž je aplikace určena primárně.

Od techniků byl na základě ústního rozhovoru zjištěn vcelku kladný ohlas. Aplikace je dle jejich názoru velice jednoduchá na instalaci a automatická konfigurace po

prvním spuštění nevyžaduje žádný uživatelský zásah. Největší přínos vidí v diagnostické části aplikace a plně automatickém zjištění stavu sítě, které uživatel pouze přečte nebo zašle emailem v případě funkčního připojení. U velké části telefonický rozhovorů tak odpadá čas strávený vysvětlováním postupů manuální diagnostiky. Celkově hodnotí aplikaci jako přínosnou a uvádějí nápady na možné inovace, mezi které patří například již zmíněné dodělání kompletního propojení na okolní monitorovací systémy, vylepšení a zjednodušení vytváření výjimky pro bránu firewall nebo zapracování zmíněného mechanismu pro obejití sítě schované za NATem.

Z řad vybraných koncových uživatelů byly formou ústní debaty zjištěny většinou kladné ohlasy. Jako největší přínos hodnotí samotný nápad a realizaci takové podpůrné aplikace, která jim usnadní komunikaci s technickou podporou. Dále pak velice kladně hodnotí zkrácení doby potřebné k vyřízení telefonátu zabývajícího se reportingem stavu linky, který mohou využít ve svůj jiný osobní prospěch. Instalace aplikace není dle jejich názoru nijak složitá a samotná orientace v aplikaci a její ovládání také ne. Záporně naopak hodnotili jednoduchý a strohý vzhled celé aplikace včetně tooltip nabídek na hlavním panelu. Jejich hodnocení aplikace bylo dle očekávání spíše netechnického rázu.

V současné době byla aplikace nabídnuta pro testování v širším okruhu uživatelů a čeká se na zpětnou vazbu.

9 Instalace a údržba

9.1 Instalace serverové části

Před samotnou instalací serverové části je nutno zajistit prostředí vhodné pro běh tohoto řešení. Situaci zjednodušuje fakt, že webová aplikace je naprogramována v PHP a dává tímto administrátorovi volnost ve výběru operačního systému serveru. Samotný script není nikterak náročný a dostatečně poslouží jakýkoliv hardware, který je schopen provozovat databázi MySQL. Na severu je nutno zajistit tyto služby:

- Apache
- PHP
- MySQL

Dále je potřeba zajistit přítomnost databáze pro ukládání provozních dat řešení, pro snadnější identifikaci je pojmenována „hkfmondb“. Tabulky lze vytvořit ručně za použití následujícího klíče.

- technics
    - ID - mediumint(9) NOT NULL AUTO_INCREMENT
    - name - varchar(30) DEFAULT NULL
    - phone - varchar(16) DEFAULT NULL
    - email - varchar(30) DEFAULT NULL
    - note - varchar(255) DEFAULT NULL
- clients
    - ip - varchar(15) DEFAULT NULL
- messages
    - ID - mediumint(9) NOT NULL AUTO_INCREMENT
    - subnet - varchar(15) DEFAULT NULL
    - datum - date DEFAULT NULL
    - parrent - int(11) DEFAULT NULL
    - text - varchar(255) DEFAULT NULL

Systémovým řešením je použití scriptu „hkfmondb.sql“ pro dávkové vygenerování tabulek a počátečního obsahu. Všechny potřebné komponenty pro nasazení serverové části jsou umístěny na adrese:

http://lide.hkfree.org/~creative/hkfreemonitor/instalace/

nebo přiloženy na médiu. Tento script již obsahuje záznam o aktuálních pracovnících linky technické podpory. Jeho spuštění se provede například, přihlášením do mysql obslužného prostředí a použitím příslušných příkazů.

- mysql> create database hkfmondb;
- mysql> use hkfmondb;
- mysql> source ~/ hkfmondb.sql;

Následuje již samotné umístění stěžejních PHP souborů tam, kde budou dostupné přes webové služby klientským aplikacím. Jejich adresa je v aplikaci nastavitelná. Jedná se o soubory:

- hkfmon.php
- index.php
- config.php

První z nich zajišťuje obsluhu klientské aplikace, a sice vrací IP adresu v síti HKfree. Generuje kompletní konfigurační soubor, vyhledává a odesílá dostupné zprávy na serveru atp. Index.php pak slouží jako administrační rozhraní a zobrazuje tabulky uložených zpráv, adres klientských aplikací a seznam pracovníků technické podpory. Výše uvedené údaje lze mazat i zakládat. V posledním souboru jsou uvedeny přístupy do databáze a její název. Tyto údaje jsou potřeba editovat dle nastavení serveru.

```
<?
$dbhost = '10.107.179.1';
$dbuser = 'kaspi';
$dbpass = '917800917800';
$dbname = "test";
?>
```

Z důvodu bezpečnosti není doporučeno ukládat tyto soubory do stejného adresáře, ale rozdělit je do veřejně dostupného adresáře pro obslužné scripty (hkfmon.php, index.php) a veřejně nedostupného adresáře pro konfigurační scripty (index.php).

Nakonec je nutno vytvořit adresář pro ukládání nových aktualizací klientské aplikace, který bude opět dostupný přes webové služby a budou z něj stahovány manuální či automatické aktualizace. Jeho počáteční obsah je možno nalézt opět na již zmíněné internetové adrese s instalací nebo na přiloženém médiu.

9.2 Instalace klientské části

Pro instalaci klientského programu byl vytvořen jednoduchý instalátor zabezpečující uživatelsky přívětivé zavedení aplikace do systému, dostupný na těchto místech:

- přiložené médium, adresář instalace
- internetová stránka:

http://lide.hkfree.org/~creative/hkfreemonitor/instalace/HKFmon_setup.exe

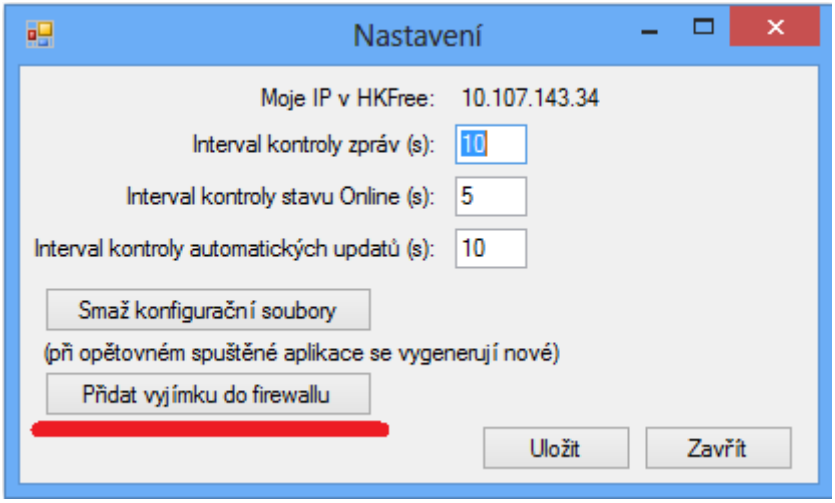
Po stažení a spuštění instalátoru stačí projít celým procesem bez změn přednastavených hodnot, po dokončení procesu se aplikace sama spustí. Při prvním spuštění bude nahlášena absence konfiguračních souborů, které budou následně automaticky vygenerovány a uloženy do adresáře k aplikaci.

Umístění programu je cíleně voleno do datového adresáře „AppData/Roaming“ aktivního uživatelského účtu počítače, který je určen právě pro bezproblémové umístění uživatelských dat. Pokud dojde při instalaci ke změně adresáře na jiný, s menším oprávněním k provádění souborových operací, může to mít za následek znemožnění ukládání konfiguračních souborů a provádění automatických aktualizací.

V případě zapnuté Brány Windows Firewall bude při generování konfiguračních souborů na pozadí spuštěn proces, zajišťující pokus o vyvolání detekce síťového

provozu a uložení výjimky do firewallu, kterou je pro správný běh aplikace nutno kladně vyřídit. Tato nabídka se zobrazí v závislosti na verzi operačního systému a nastavení firewallu. V případě neúspěšnosti je nutno podniknout následující kroky k uložení výjimky ručně.

Při vkládání výjimky do firewallu pomocí aplikace je nutno, aby byla spuštěna s oprávněními správce, poté stačí zvolit v okně nastavení volbu „Přidat výjimku do firewallu“, vše bude nastaveno automaticky, a bude zobrazena správa o úspěšnosti či neúspěšnosti operace.



Obrázek 15 Okno Nastavení klientské aplikace
Zdroj: Autor

Manuální vložení může být provedeno přímo v nastavení Brány Windows Firewall přidáním pravidla. Zde je doporučeno přidat výjimku pro celou aplikaci, nikoliv pro určité komunikační porty, ty jsou nastavitelné a jejich změna by mohla vést k nefunkční komunikaci mezi jednotlivými klienty.

9.3 Údržba

Údržba serverové části

Jednou za delší časové období (rok a více) bude vhodné smazat staré zprávy pomocí běžných databázových dotazů, a jinak kromě běžných úkonů údržby databázové serveru, není potřeba provádět žádné jiné zvláštní operace spojené s přítomností tohoto řešení.

Údržba klientské části

Aplikace je naprogramována tak, aby žádnou pravidelnou údržbu nebylo nutné provádět. Jediná ukládaná data jsou předávané zprávy a konfigurace programu, které ani za několik let nedokáží zabrat na disku počítače tolik místa, aby stálo za úvahu podstoupit opravné kroky. Je ale možné, že aplikace bude postupem času agregovat takové množství přijatých zpráv, že pro uživatele nebude potřebné je ukládat. V nastavení je proto zabudována funkce smazání konfiguračních souborů a po restartu dojde k jejich opětovnému vygenerování s tím, že bude staženo pouze přiměřené množství zpráv.

10 Závěr

Díky této práci se podařilo vyvinout diagnostický aparát pro spolek HKfree ve formě klientské aplikace a její serverové podpůrné části. Klientská aplikace byla naprogramována pomocí jazyka C# ve vývojové prostředí programu Visual Studio 2010. Tímto je určena pro klientské stanice s operačním systémem Windows a řeší otázky jednoduché diagnostiky síťového připojení a informovanosti o aktuálním stavu sítě. Diagnostika je prováděna převážně pomocí údajů navrácených použitím Ping a Tracert v kombinaci se znalostí topologie sítě a hodnotami uloženými v serverové části. Rozesílání zpráv o aktuálním stavu sítě je v případě dostupnosti realizováno vzájemnou komunikací se serverovou částí a v opačném případě peer-to-peer komunikací mezi klientskými aplikacemi. Serverová část byla naprogramována v jazyce PHP, jako úložiště dat využívá databázi MySQL. Serverová část slouží jako úložiště zasílaných zpráv, seznamu adres klientských stanic a seznamu pracovníků technické podpory. Pro techniky dále zajišťuje možnost editace těchto hodnot přes webové rozhraní.

Systém byl jako celek podroben testování v různých prostředích a byly odladěny počáteční chyby tak, aby byl provozuschopný v reálném prostředí. Tímto byly splněny cíle práce v plném rozsahu, což bylo ověřeno ústní zpětnou vazbou od vybraných jedinců, kteří aplikaci testovali. V současné době byla aplikace postoupena jak pracovníků linky technické podpory tak běžným uživatelům pro běžný provoz, který dále ukáže nové možnosti inovací či oprav.

Autorem navrhované konkrétní náměty k rozšiřování stávajících funkcí aplikace byly postupně vytyčeny v příslušných kapitolách práce. Zcela nové inovace by se mohly ubírat směrem moderního vzhledu, který by ještě více přiblížil koncovou aplikaci běžným uživatelům. Dále by mohla být aplikace napojena na stávající informační systém, který spravuje kompletní agendu všech aktivních členů sítě HKfree za účelem zobrazování a aktualizací osobních údajů.

11 Seznam použité literatury

[1] BIGELOW, Stephen J. Mistrovství v počítačových sítích: správa, konfigurace, diagnostika a řešení problémů. Vyd. 1. Překlad Petr Matějů. Brno: Computer Press, 2004, 990 s. ISBN 80-251-0178-9.

[2] DOSTÁLEK, Libor. Velký průvodce protokoly TCP/IP a systémem DNS. 3. aktualiz. a rozš. vyd. Praha: Computer Press, 2002, xiv, 542 s. ISBN 80-722-6675-6.

[3] HORÁK, Jaroslav a Milan KERŠLÁGER. Počítačové sítě pro začínající správce. 3., aktualiz. vyd. Brno: Computer Press, 2006, 211 s. ISBN 80-251-0892-9.

[4] KOUTNÁ, Marcela a Tomáš SOCHOR. Úvod do počítačových sítí. Orlová, 2006. Dostupné z: http://distancne.obaka-orlova.cz/PDF/UPS.pdf. Učební text. Obchodní akademie Orlová.

[5] KRČMÁŘ, Petr. Přelez, přeskoč a podlez NAT. In: ROOT.CZ [online]. 2006 [cit. 2014-11-03]. Dostupné z: http://www.root.cz/clanky/prelez-preskoc-a-podlez-nat/

[6] Proč není NAT totéž co firewall. In: KRČMÁŘ, Petr. ROOT.CZ [online]. 2007 [cit. 2014-11-02]. Dostupné z: http://www.root.cz/clanky/proc-neni-nat-totez-co-firewall/

[7] LOMNICKÝ, Marek a Vladimír VESELÝ. Směrování a směrovací protokoly. Brno, 2007. Dostupné z: http://netacad.fit.vutbr.cz/texty/ccna-moduly/ccna2-6.pdf. Učební text. Vysoké učení technické v Brně.

[8] LONGER. Co je to ping a jak ho zjistím?. Pcblog.cz [online]. 2008, č. 1317 [cit. 2014-11-02]. Dostupné z: http://www.pcblog.cz/clanek/1317/co-je-ping-k-cemu-slouzi-jak-na-ping/

[9] OSPF. MICROSOFT. OSPF [online]. 2014 [cit. 2014-11-02]. Dostupné z: http://technet.microsoft.com/cs-cz/library/cc778874(v=ws.10).aspx

[10] PŘÍHODA, Petr. Počítačové sítě. Olomouc, 2007. Dostupné z: http://phoenix.inf.upol.cz/esf/ucebni/poc_site.pdf. Učební text. Univerzita Palackého.

[11] SCHOVÁNEK, Jan. *Nástroj pro návrh a dohled OSPF*. Hradec Králové, 2013. Diplomová práce. Univerzita Hradec Králové. Vedoucí práce Ing., Pavel Kříž, Ph.D.

12 Seznam obrázků

13 Seznam tabulek

14 Přílohy

1) TCP spojení (zdrojový kód) – klientská část
2) Přiložené CD s obsahem:
    a) Instalace
    b) Visual Studio 2010 projekty

TCP spojení (zdrojový kód) – klientská část

```
internal string getListSync(string command, string address)
        {
            try
            {
                Socket soc = new Socket(AddressFamily.InterNetwork,
SocketType.Stream, ProtocolType.Tcp);
                soc.SetSocketOption(SocketOptionLevel.Socket,
SocketOptionName.SendTimeout, 5000);
                soc.SetSocketOption(SocketOptionLevel.Socket,
SocketOptionName.ReceiveTimeout, 5000);

                IAsyncResult result =
soc.BeginConnect(IPAddress.Parse(address), conf.serverPort, null,
null);

                bool success = result.AsyncWaitHandle.WaitOne(5000,
true);
                if (!soc.Connected)
                {
                    return null;
                }

                byte[] buffer =
Encoding.GetEncoding(1250).GetBytes(command);
                soc.Send(buffer);
                byte[] incoming = new byte[262144];
                int incLength = soc.Receive(incoming);
                string message =
Encoding.GetEncoding(1250).GetString(incoming, 0, incLength);
                System.Threading.Thread.Sleep(30);

                while (soc.Available > 0)
                {

                    incLength = soc.Receive(incoming);
                    message +=
Encoding.GetEncoding(1250).GetString(incoming, 0, incLength);
                    System.Threading.Thread.Sleep(50);
                }
                soc.Close();
                string response = message;

                string tmp = response;
                if (tmp == "-;-;") tmp = ";";
                recComplete = false;
                response = "";
                if (tmp.EndsWith(";")) tmp = tmp.Substring(0,
tmp.Length - 1);
                return tmp;
            }
            catch (Exception)
            {
                return null;
            }
        }
```

Oskenované zadání práce

UNIVERZITA HRADEC KRÁLOVÉ
Fakulta informatiky a managementu
Rokitanského 62, 500 03 Hradec Králové, tel: 493 331 111, fax: 493 332 235

Zadání k závěrečné práci

Jméno a příjmení studenta: **Martin Kašpar**

Obor studia: Informační management (5)
Jméno a příjmení vedoucího práce: **Pavel Kříž**

Název práce:
Distribuovaný informační a diagnostický systém sítě

Název práce v AJ:
Distributed informational and diagnostic network system

Podtitul práce:

Podtitul práce v AJ:

Cíl práce: Analýza současného stavu, návrh a implementace aplikace pro diagnostiku a informovanost koncových uživatelů sítě HKfree.

Osnova práce:
1. Úvod
2. Cíl práce
3. Teorie počítačových sítí
4. Analýza prostředí a stavu
5. Návrh řešení
6. Implementace
7. Testování
8. Instalace a údržba
9. Závěr

Projednáno dne: 14.10.2013

Podpis studenta

Podpis vedoucího práce