

Univerzita Pardubice

Fakulta elektrotechniky a informatiky

Mapování sítě Internet pomocí paketů ICMP

Luboš Čábelka

Diplomová práce

2015

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2014/2015

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Luboš Čábelka**
Osobní číslo: **I13400**
Studijní program: **N2646 Informační technologie**
Studijní obor: **Informační technologie**
Název tématu: **Mapování sítě internet pomocí paketů ICMP**
Zadávací katedra: **Katedra softwarových technologií**

Z á s a d y p r o v y p r a c o v á n í :

Cílem práce je návrh a implementace aplikace pro zachytávání paketů ICMP a tvorbu mapy Internetových zařízení v jazyce Java. V teoretické části bude popis současných technologií pro mapování Internetu a map Internetových zařízení. Dále popis důležitých síťových zařízení, zejména routerů. Implementací bude Java GUI aplikace, která bude vysílat a zachytávat pakety. Pro ukládání paketů a výsledků mapování bude použita jednoduchá databázová aplikace. Výsledky se budou vykreslovat na obrazovku.

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

PECINOVSKÝ, Rudolf. Návrhové vzory: návrh a tvorba aplikací. Vyd. 1. Brno: Computer Press, 2008, 527 s. Programování. ISBN 978-80-251-1582-4.

JONES, Meilir. Základy objektově orientovaného návrhu v UML: návrh a tvorba aplikací. Vyd. 1. Praha: Grada, 2001, 367 s. Programování. ISBN 80-247-0210-X.

PUŽMANOVÁ, Rita. TCP/IP v kostce: návrh a tvorba aplikací. 1. vyd. České Budějovice: Kopp, 2004, 607 s. Programování. ISBN 80-723-2236-2.

Vedoucí diplomové práce:

Ing. Zdeněk Šilar, Ph.D.

Katedra informačních technologií

Datum zadání diplomové práce:

31. října 2014

Termín odevzdání diplomové práce:

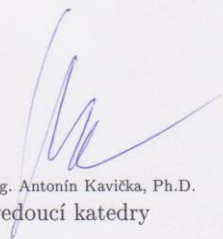
15. května 2015



prof. Ing. Simeon Karamazov, Dr.
děkan



L.S.



prof. Ing. Antonín Kavička, Ph.D.
vedoucí katedry

V Pardubicích dne 15. listopadu 2014

Prohlašuji:

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Kojicích dne 14.5.2015

Luboš Čábelka

Poděkování

Na tomto místě bych chtěl poděkovat Ing. Zděnkovi Šilarovi Ph.D za vedení mé diplomové práce a podporu při studiu.

ANOTACE

Práce je věnována zachytávání a odesílání paketů (zejména ICMP echo) v jazyce Java, dále tvorbě mapy internetových zařízení, především směrovačů a jejich rozhraní. V praktické části je využito představených technologií pro tvorbu mapy internetových uzlů.

KLÍČOVÁ SLOVA

Java, síť, ICMP, mapa sítě

TITLE

Creating Internet map using ICMP packets

ANNOTATION

The work deals with capturing and sending packets (ICMP echo) in programming language Java. Deals with network maps, devices, mostly routers and its interfaces. Described technologies are used in practical part for sending and capturing packets, and creating Internet map.

KEYWORDS

Java, network, ICMP, Internet map

Obsah

Úvod.....	13
Teoretická část	14
1. Síťové protokoly	14
1.1 Protokol ICMP	14
1.2 Protokol IP verze 4.....	17
1.3 Protokol IP verze 6.....	20
1.4 DNS.....	21
2. Mapy Internetu.....	22
2.1 Výběr map Internetu	22
2.1.1 Arpanet	23
2.1.2 Greg's Cable Map.....	24
2.1.3 Internetová mapa provozu na webových stránkách.....	25
2.1.4 Internet Mapping Project.....	26
2.1.5 The Opte Project.....	28
2.2 Databáze IP adres	29
2.2.1 ICANN	30
2.2.2 Databáze IP adres www.czdomeny.cz	31
2.2.3 Databáze IP adres https://db-ip.com/	32
3. Síťová zařízení	33
3.1 Hub (opakovač).....	33
3.2 Switch a Bridge (Přepínač a most – dvouportový přepínač).....	33
3.3 Router (Směrovač)	34
4. Java technologie.....	35
4.1 Knihovny WinPcap a Jpcap.....	35
4.1.1 Třídy knihovny Jpcap	35
4.2 Java API pro tvorbu GUI	36

4.2.1 JavaFX	36
Praktická část	37
5. Návrh aplikace	37
5.1 Základ aplikace	37
5.2 Návrh databáze	38
5.2.1 Model databáze	38
6. Vývoj aplikace	40
6.1 Úpravy databáze	40
6.1.1 Odebrání tabulky trasy	40
6.1.2 Změna tabulky sousedé na předchůdci	40
6.1.3 Přidání sloupců pro polohu na mapě	40
6.1.4 Odebrání tabulky pakety	40
6.2 Grafické rozhraní	41
6.2.1 Podpora v Netbeans	41
7. Struktura aplikace	42
7.0.1 Instalace vývojového prostředí	42
7.1 Struktura Aplikace	42
7.2 Databáze	43
7.3. Vstupy aplikace	46
7.3.1 Tvorba trasy	46
7.3.2 Načtení ze souboru	47
7.3.3 Ruční editace databáze	48
7.4 Uložení trasy do databáze	48
7.5 Uložení paketů do souboru	50
7.6 Ovládání aplikace	50
7.6.1 Virtuální mapa rozhraní	51
7.6.2 Editace databáze	52

7.6.3 Zachytávač paketů	53
7.6.4 Reálná mapa rozhraní	54
8. Vytvořené mapy.....	55
8.1 Virtuální mapa rozhraní	55
8.2 Mapa rozhraní s reálným mapovým podkladem	56
Závěr	57
Literatura.....	59
Příloha A - Obsah CD	61
Příloha B – Virtuální mapa rozhraní	62
Příloha C – Reálná mapa rozhraní	67
Příloha D – Stěhování serverů firmy seznam.cz	68

Seznam obrázků

Obrázek 1 Formát ICMP echo zprávy[3]	15
Obrázek 2 Ukázka použití příkazu ping ve Windows	16
Obrázek 3 Ukázka použití příkazu traceroute ve Windows	16
Obrázek 4 Formát zprávy ICMP destination unreachable[3]	17
Obrázek 5 Hlavička IPv4 datagramu[3]	18
Obrázek 6 Hlavička IPv6[3]	20
Obrázek 7 Síť Arpanet a sítě k ní připojené[17]	23
Obrázek 8 Rozšíření sítě v USA v letech 1969-1977[16]	24
Obrázek 9 Mapa podmořských kabelů[9]	25
Obrázek 10 Pohled na celkovou mapu provozu[18]	25
Obrázek 11 Míra provozu v jednotlivých zemích[18]	26
Obrázek 12 Mapa hlavních poskytovatelů[10]	27
Obrázek 13 Mapa firmy Lumeta[11]	28
Obrázek 14 Mapa Internetu projektu OPTE[20]	29
Obrázek 15 Databáze IPv4 rozsahů adres[21]	30
Obrázek 16 Reverzní záznam IP adresy[12]	31
Obrázek 17 Záznam pro www.microsoft.com[19]	32
Obrázek 18 Okno aplikace pro zachytávání a analýzu paketů	38
Obrázek 19 Původní návrh databáze vytvořený v rámci semestrální práce	39
Obrázek 20 Ukázka okna aplikace pro ovládání databáze	39
Obrázek 21 Ukázka návrhu aplikace v JavaFX Scene Builder	41
Obrázek 22 Třídy projektu diplomové práce	43
Obrázek 23 Finální model databáze	45
Obrázek 24 Ukázka okna aplikace se zvolenou virtuální mapou	51
Obrázek 25 Ukázka okna aplikace se zvolenou editací databáze	52
Obrázek 26 Výběr síťového rozhraní pro zachytávání	53
Obrázek 27 Ukázka okna aplikace se zachytáváním paketů	53
Obrázek 28 Ukázka okna aplikace s reálným mapovým podkladem	54
Obrázek 29 Virtuální mapa rozhraní	55
Obrázek 30 Reálná mapa rozhraní	56
Obrázek 31 Rozložení listů virtuální mapy	62
Obrázek 32 Mapa trasy k serverům seznam.cz ze září 2014	68

Obrázek 33 Mapa trasy k serverům seznam.cz z dubna 2015	69
Obrázek 34 Mapa trasy k serverům seznam.cz z 12. května 2015	69

Seznam tabulek

Tabulka 1 Nejčastější typy ICMP zpráv	15
Tabulka 2 Význam sloupců tabulky <i>rozhrani</i>	44
Tabulka 3 Význam sloupců tabulky <i>predchudci</i>	44
Tabulka 4 Význam sloupců tabulky <i>adresy</i>	44
Tabulka 5 Význam sloupců tabulky <i>smerovace</i>	45

Seznam zkratek

ICMP – Internet Control Message Protocol

IP – Internet Protocol

RFC – Request for Comments

UDP – User Datagram Protocol

TTL – Time To Live

TCP – Transmission Control Protocol

IPv4 – Internet Protocol verze 4

IPv6 - Internet Protocol verze 6

IANA – Internet Assigned Number Authority

ICANN – Internet Corporation for Assigned Names and Numbers

IPsec – Internet Protocol security

DNS – Domain Name System

DDos – Distributed Denial of Service

CAT5e – Category 5e

JPCAP – Java Packet Captor

BLOB – Binary Large Object

API – Application Programming Interface

MVC – Model View Controller

FXML – FX Markup Language

MTU – Maximum Transmission Unit

DHCP – Dynamic Host Configuration Protocol

Úvod

Jako svou diplomovou práci jsem zvolil téma Mapování sítě Internet pomocí paketů ICMP. Navazuji na svou bakalářskou práci, kde jsem se zabýval zachytáváním a analýzou síťových paketů.

Mapování Internetu (ve smyslu tvorby mapy, ne převod například adres z jednoho typu na jiný) se může provádět různými způsoby, asi nejjednodušší je trasování (traceroute, tracert) k cíli a ukládání výsledku, nebo se může využít vyhledávání, kudy jde které spojení na Internetu, jako například mapa podmořských kabelů[9].

Práce je rozdělena na dvě hlavní části. Teoretická část popisuje síťové protokoly, síťová zařízení a vybrané mapy Internetu. Praktická část se zabývá návrhem a implementací aplikace pro odesílání a zachytávání paketů, ukládání zjištěných informací do databáze a jejich vykreslování na obrazovku v podobě map, dále vytvořením dvou vlastních map, jedné virtuální a jedné fyzické, která odráží reálnou polohu uzlů v síti a vykresluje je na mapu.

Jazykové konvence

Pro zvýšení čitelnosti textu byla nastavena následující jednotná konvence zápisu textu.

Důležitý text	podstatné texty, důležité názvy, zkratky
<u>zdrojový kód</u>	zdrojové kódy aplikace
<i>název třídy</i>	názvy tříd, metod, proměnných atd.

Teoretická část

1. Síťové protokoly

Síťový protokol je standard nebo doporučení, podle kterého spolu komunikují zařízení po síti. Definuje pravidla, formát paketů a procedur pro výměnu informací mezi dvěma komunikujícími prvky.

1.1 Protokol ICMP

Internet Control Message Protocol. Protokol síťové vrstvy určený k přenosu zpráv o chybách a zvláštních okolnostech přenosu. Většina zpráv neslouží pro uživatele ale pro **IP** software, operační systémy, routery. **ICMP** využívá služeb **IP** v rámci síťové vrstvy. Nejčastější zprávy: oznámení o nedostupnosti služby, zahození paketu, nedostupnosti sítě. **ICMP** protokol je definovaný v **RFC 792**.

1.1.1 Obsah ICMP hlavičky

Typ zprávy – specifikuje typ **ICMP** zprávy

Kód zprávy – určuje parametry zprávy

Kontrolní součet – zabezpečení záhlaví proti chybám

V tabulce 1 jsou nejčastější typy ICMP zpráv, mezi nejdůležitější zprávy patří Echo request (8, 0), Echo reply (0, 0) využívané příkazem ping a Time Exceeded (11, 0) využívané příkazem traceroute pro sestavení trasy.

Tabulka 1 Nejčastější typy ICMP zpráv

Typ hlášení	Kód zprávy	
0	0	Echo reply - odpověď na ping
8	0	Echo request - žádost o ping
3	0	Nedostupná síť
3	1	Nedostupná stanice
3	2	Nedostupný protokol
3	3	Nedostupný port
3	4	Nemožnost fragmentace když byla potřeba
3	5	Nevydařené směrování
4	0	Informace o zahltění
5	0	Přesměrování datagramů pro síť
5	1	Přesměrování datagramů pro stanici
5	2	Přesměrování datagramů pro typ služby a síť
5	3	Přesměrování datagramů pro typ služby a stanici
11	0	TTL vypršelo
11	1	Vypršel čas znovusestavení datagramu
13	0	žádost o časové razítko
14	0	odpověď s časovým razítkem
15	0	žádost o adresu sítě
16	0	Odpověď s adresou sítě

1.1.2 Ping (Packet Internet Groper)

Paket ICMP ping slouží k ověření dostupnosti stanice nebo zařízení. Pracuje na nejnižší možné vrstvě, je implementován přímo v operačním systému (ve Windows příkaz ping). Stanice vyšle zprávu **ICMP** echo request a očekává od cíle odpověď **ICMP** echo reply, od doby odeslání se měří čas do doby přijetí odpovědi. Na obrázku 1 je formát ICMP echo zprávy. Na obrázku 2 je využití příkazu ping v příkazovém řádku ve Windows.

typ = 8 request typ = 0 reply	kód = 0	Kontrolní součet
Identifikátor		Pořadové číslo
Volitelná data		

Obrázek 1 Formát ICMP echo zprávy[3]

```

C:\Users\Čábelka>ping www.seznam.cz

Pinging www.seznam.cz [77.75.76.3] with 32 bytes of data:
Reply from 77.75.76.3: bytes=32 time=6ms TTL=250
Reply from 77.75.76.3: bytes=32 time=5ms TTL=250
Reply from 77.75.76.3: bytes=32 time=5ms TTL=250
Reply from 77.75.76.3: bytes=32 time=8ms TTL=250

Ping statistics for 77.75.76.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 5ms, Maximum = 8ms, Average = 6ms

```

Obrázek 2 Ukázka použití příkazu ping ve Windows

1.1.3 Traceroute

Trasování cesty slouží ke zjištění všech směrovačů v cestě k cíli. Systém Windows obvykle vytváří datagramy **ICMP**, unixové systémy používají **UDP**, ale může být i **TCP**. Vygeneruje se datagram (některé systémy vysílají několikrát, například Windows odesílá každý datagram 3x) s hodnotou **TTL** 1 a je odeslán k cíli (ten je na prvním směrovači zahozen a směrovač vyšle zpět zprávu **ICMP** Time exceeded – typ = 11, kód = 0), poté se generuje stejný datagram s postupně se zvyšujícím polem **TTL**, až je dosaženo cíle. Z odpovědí od směrovačů po cestě lze sestavit cestu k cíli. Na obrázku 3 je ukázka příkazu traceroute (tracert) ve Windows.

```

C:\Users\Čábelka>tracert www.seznam.cz

Tracing route to www.seznam.cz [77.75.76.3]
over a maximum of 30 hops:

  1     1 ms    <1 ms    <1 ms   192.168.1.100
  2     2 ms     1 ms     1 ms   10.11.2.254
  3     3 ms     2 ms     2 ms   10.11.2.149
  4     5 ms     7 ms     4 ms   95.85.240.89
  5     6 ms     5 ms     5 ms   nix2.seznam.cz [91.210.16.194]
  6     5 ms     5 ms     5 ms   www.seznam.cz [77.75.76.3]

Trace complete.

```

Obrázek 3 Ukázka použití příkazu traceroute ve Windows

1.1.4 ICMP Destination Unreachable

V případě nedostupnosti stanice (sítě, protokolu, portu, atd.) je zpět odeslán datagram **ICMP** s informací o nedostupnosti a součástí tohoto datagramu je zpět odeslána původní hlavička a prvních 64 bitů dat. Tato data mohou sloužit jako identifikace programu, který právě komunikuje (pokud by například jedna aplikace prováděla ping a druhá trasování na stejnou adresu, datagramy by se pomíchaly a trasování by skončilo ihned po přijetí datagramu **ICMP** echo reply, což by program provádějící trasování považoval za odpověď od cíle a ukončil by trasování). Na obrázku 4 je formát zprávy **ICMP** destination unreachable

typ = 3	kód = 0-12	Kontrolní součet
0		
Záhlaví původního datagramu + prvních 64 bitů dat		

Obrázek 4 Formát zprávy ICMP destination unreachable[3]

1.2 Protokol IP verze 4

Protokol **IP** vysílá datagramy dle síťových adres, nenavazuje spojení, ani neuchovává informace o odeslaných datagramech. Nekontroluje doručení datagramů, o to se musí postarat protokoly vyšších vrstev.

Adresa **IPv4** má 32 bitů, maximálně tedy 4 294 967 296 adres. Ne všechny se však mohou použít pro koncové stanice, protože některé adresy jsou adresy sítě a adresy pro všesměrové vysílání (broadcast). Příklad **IP** adresy **192.168.1.1**. O přidělování adres se původně starala společnost **IANA** (Internet Assigned Numbers Authority) přidělovala rozsahy **IP** adres jednotlivým národním poskytovatelům a ti je distribuovali dále, dnes tuto činnost vykonává korporace **ICANN** (Internet Corporation for Assigned Names and Numbers). Spravuje kořenové **DNS** servery, **IP** adresy a registry protokolů.

Třídy adres

- **Třída A** – 126 adres po $2^{24} - 2$ uzlů. Největší síť s nejmenším počtem uzlů. Pro adresu sítě se používá prvních 8bitů. Adresy sítí 1.0.0.0 - 126.0.0.0
- **Třída B** – 2^{14} adres po $2^{16} - 2$ uzlů. Středně velké sítě se středním počtem uzlů, Pro adresu sítě se používá prvních 16bitů. Adresy sítí 128.1.0.0 - 191.255.0.0
- **Třída C** – 2^{21} adres po $2^8 - 2$ uzlů. Hodně malých sítí s malým počtem uzlů. Pro adresu sítě se používá prvních 24bitů. Adresy sítí 192.0.0.0 - 223.255.255.0

- **Třída D** – pro skupinové vysílání, adresy: 224.0.0.0 - 239.255.255.255
- **Třída E** – slouží pouze pro experimentální účely a jako rezerva, adresy 240.x.x.x. - 254.x.x.x

Používání adres třídy A v počátcích vedlo k velkému plýtvání adresami. Například některé univerzity a organizace, které se jako první připojovaly k síti Arpanet, mají přidělené adresy třídy A s 16 miliony **IP** adres.

Privátní adresy

V prvních třech skupinách jsou definované privátní adresy pro stanice skryté za směrovačem, který působí jako brána do internetu. Pakety s privátní zdrojovou nebo cílovou adresou nejsou směrovány v síti, ale pokud nějaký omylem projde do internetu je zahozen. Privátní adresy se mohou opakovat (v každé privátní síti jednou, v celém Internetu mnohokrát).

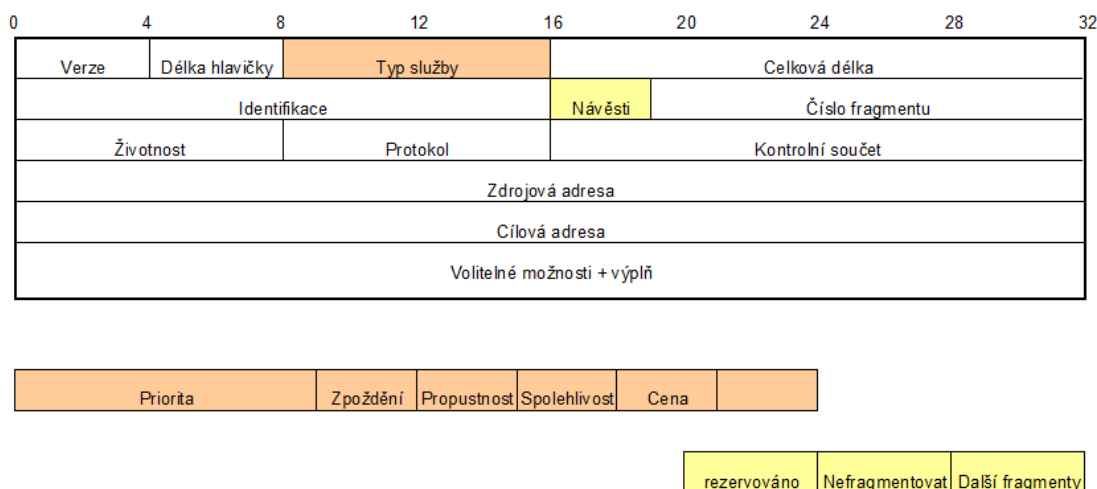
Třída A – privátní adresa 10.0.0.0

Třída B – privátní adresy 172.16.0.0 - 172.31.0.0

Třída C – privátní adresy 192.168.0.0 - 192.168.255.0

IPv4 Datagram

Hlavička **IP** datagramu verze 4 má délku 20 bytů + volitelné možnosti. Na obrázku 5 je hlavička datagramu.



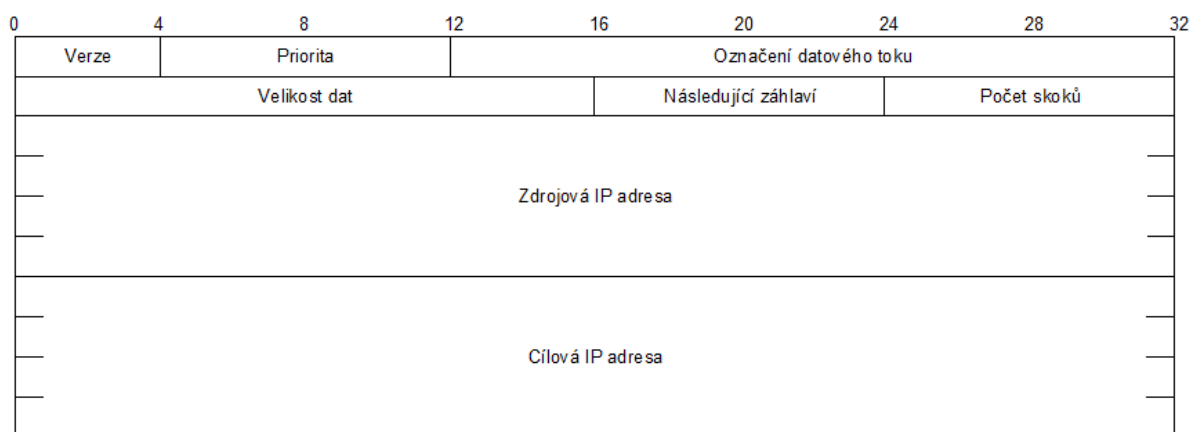
Obrázek 5 Hlavička IPv4 datagramu[3]

Obsah hlavičky IP verze 4

- **Verze** (Version, 4bity) – verze IP protokolu v IPv4 vždy 4
- **Délka hlavičky** (Internet Header Length, 4bity) – Délka hlavičky v násobcích 32 maximum $15 \cdot 32 = 480$ bitů
- **Typ služby** (Type of service, 8bitů) – Informace pro směrovače jak mají s datagramem zacházet, toto pole se obvykle nepoužívá
 - Priorita** – nejvyšší = 7
 - Zpoždění** – minimální zpoždění = 1
 - Propustnost** – maximální propustnost = 1
 - Spolehlivost** – maximální spolehlivost = 1
 - Cena** – minimální cena = 1
- **Celková délka** – celková velikost datagramu v násobcích 8 maximálně $65\,535 \cdot 8$
- **Identifikace** – jedinečná identifikace datagramu, využívá se pro fragmentaci
- **Návěsti** – používají se pro fragmentaci
 - Rezervováno – vždy 0
 - Nefragmentovat** – paket nemůže být cestou fragmentován (1 znamená nefragmentovat)
- **Další fragmenty** – označuje, jestli následují další fragmenty (poslední fragment = 0)
- **Číslo fragmentu** – jednoznačně určuje pořadí fragmentu jako vzdálenost od začátku datagramu v násobcích 64
- **Životnost TTL** – označuje počet směrovačů, skrz které může paket projít, než je zničen
- **Protokol** – protokol vyšší, transportní vrstvy
- **Kontrolní součet** – zabezpečuje hlavičku proti chybám (pro výpočet se nastaví hodnota 0)
- **Zdrojová adresa** – 32 bitová adresa zdroje
- **Cílová adresa** – 32 bitová adresa cíle
- **Volitelné možnosti + výplň** – zabezpečení, záznam cesty sítí, dodržení předepsané cesty + výplň na násobek 32bitů

1.3 Protokol IP verze 6

Nová verze protokolu **IP**, byla vytvořena již v roce 1995. Nový protokol byl potřeba, protože ve stávajícím protokolu **IPv4** docházely adresy (zejména kvůli jejich plýtvání v počátku). **IPv6** adresa má 128 bitů, což umožňuje vytvoření mnohem většího počtu adres. Mezi další výhody patří zjednodušení hlavičky, lepší bezpečnost díky **IPSec** (bezpečnostní protokol pro autentizaci, kontrolu integrity atd.). Na obrázku 6 je hlavička **IPv6** datagramu dlouhá 40 bajtů.



Obrázek 6 Hlavička IPv6[3]

Obsah IPv6 hlavičky

- **Verze** – verze protokolu v **IPv6** je vždy 6
- **Priorita** – priorita rychlosti přenosu, doručení
- **Označení datového toku** (flow label) – označuje datagramy, které vyžadují speciální zacházení, společně s prioritou slouží pro podporu Quality of Service
- **Velikost dat** – délka dat uložených v datagramu
- **Následující záhlaví** – označuje následující záhlaví
- **Maximální počet směrovačů** (hop limit) – označuje počet směrovačů před zahazením datagramu (obdobně jako **TTL** u **IPv4**)
- **Zdrojová adresa** – 128 bitová **IPv6** adresa zdroje
- **Cílová adresa** – 128 bitová **IPv6** adresa cíle

Protokol **IPv6** nepodporuje fragmentaci (stanice si musí zjistit maximální velikost **MTU** před odesláním datagramu), proto v záhlaví není pole pro fragmentaci, dále neobsahuje délku záhlaví, neboť záhlaví je pevné délky, ani kontrolní součet, neboť spoléhá na nižší vrstvu.

1.4 DNS

Domain Name System – Systém překladu jmen. Stará se o překlad jmen (například webová adresa `www.seznam.cz`) na **IP** adresy (`77.75.76.3`), jmenné adresy jsou pro běžného uživatele jednodušší k zapamatování, ale pro práci v síti je třeba směřovat podle **IP** adres, proto byl vytvořen protokol **DNS** pro překlad jmen na adresy a zpět.

Hiearchie domén

Kořenová úroveň – vrchol stromu doménových jmen

Vrcholová úroveň – dvoupísmenné národní domény (`cz`, `eu`, `us`, `hu`, ...), třípísmenné generické domény (`com`, `org`, `edu`, `net`, ...)

Druhá úroveň – jednotlivé názvy (`seznam`, `google`, `youtube`)

Další úrovně – řazené postupně pod předchozí úroveň

DNS zprávy

Protokol **DNS** může používat jak **TCP** tak **UDP** protokol pro přenos. Zprávy jsou dvou typů dotaz (query) a odpověď (response).

2. Mapy Internetu

Internet je celosvětová síť propojených počítačů, serverů, směrovačů a dalších zařízení. Předchůdcem Internetu byl Arpanet. Arpanet vznikl v roce 1969 jako experimentální síť s přepojováním paketů, zpočátku obsahoval 4 uzly: University of California Los Angeles, University of California Santa Barbara, University of Utah a Stanford Research Institute. Postupem času se k této síti připojovaly další uzly a sítě, z počátku především univerzity v USA.

Mapy sítě můžeme rozdělit na 2 skupiny: uzlové, které zaznamenávají každý uzel v síti a síťové, které zaznamenávají celou síť jako jeden celek. Dále je lze rozdělit na logické zaznamenávající jednotlivá spojení bez ohledu na to, kudy spojení vede fyzicky. Mohou vznikat i jako plány podle kterých je síť budována. A fyzické využívající reálné mapové podklady, názorně ukazují, kudy přesně vede dané spojení.

V současnosti prakticky neexistuje kompletní a přesná mapa celého Internetu, neboť se určité části sítě neustále mění např.: privátní adresy v sítích (nedostupné z venku), dynamicky přidělované adresy od poskytovatelů (uzel může dostat při každém připojení k síti jinou adresu), ale i dlouhodobější změny jako rušení částí sítě, vytváření nových částí, zejména v souvislosti s přechodem na protokol **IP** verze 6, dále pak překlad adres **IPv4** na **IPv6** a zpět. Navíc by mapa všech uzlů byla příliš rozsáhlá. Dle údaje společnosti Cisco je v květnu 2015 k Internetu připojeno přibližně **15,6 miliardy zařízení** (počítače, mobilní telefony, servery, kamery, specializovaná zařízení atd.). Pokud by údaje o každém rozhraní měly velikost 1 kilobajt, pak by celá databáze zabírala zhruba 15 terabajtů, což je na hranici velikosti dnes běžně dostupných pevných disků a práce s tak velkou databází by byla na běžném domácím počítači prakticky nemožná.

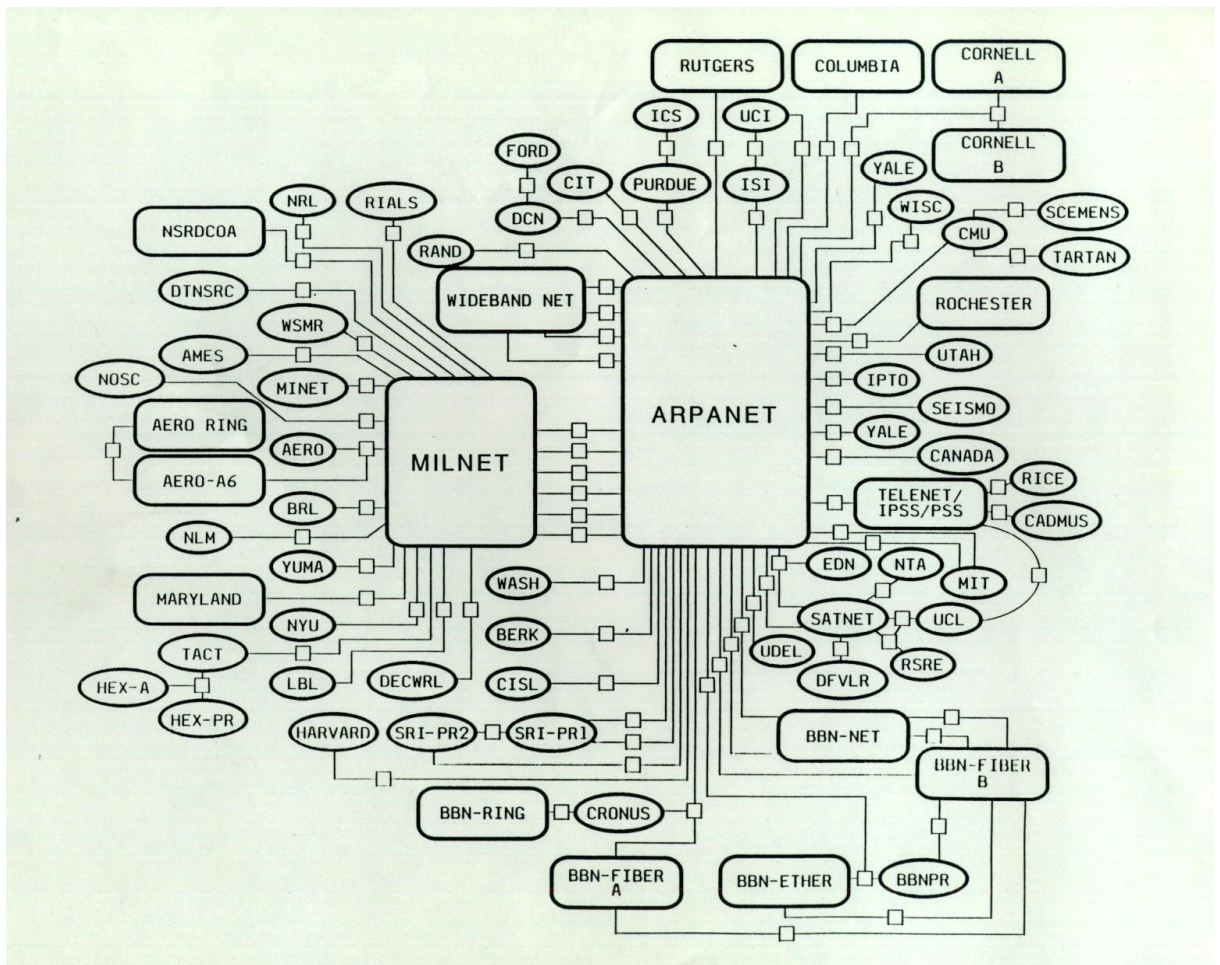
2.1 Výběr map Internetu

V současné době existuje velké množství nejrozličnějších map Internetu, ale různé mapy pokrývají různé části sítě. Za nejkompletnější mapy lze považovat původní mapy z doby Arpanetu, kdy celá síť byla na dnešní poměry velmi malá a mohla existovat kompletní mapa.

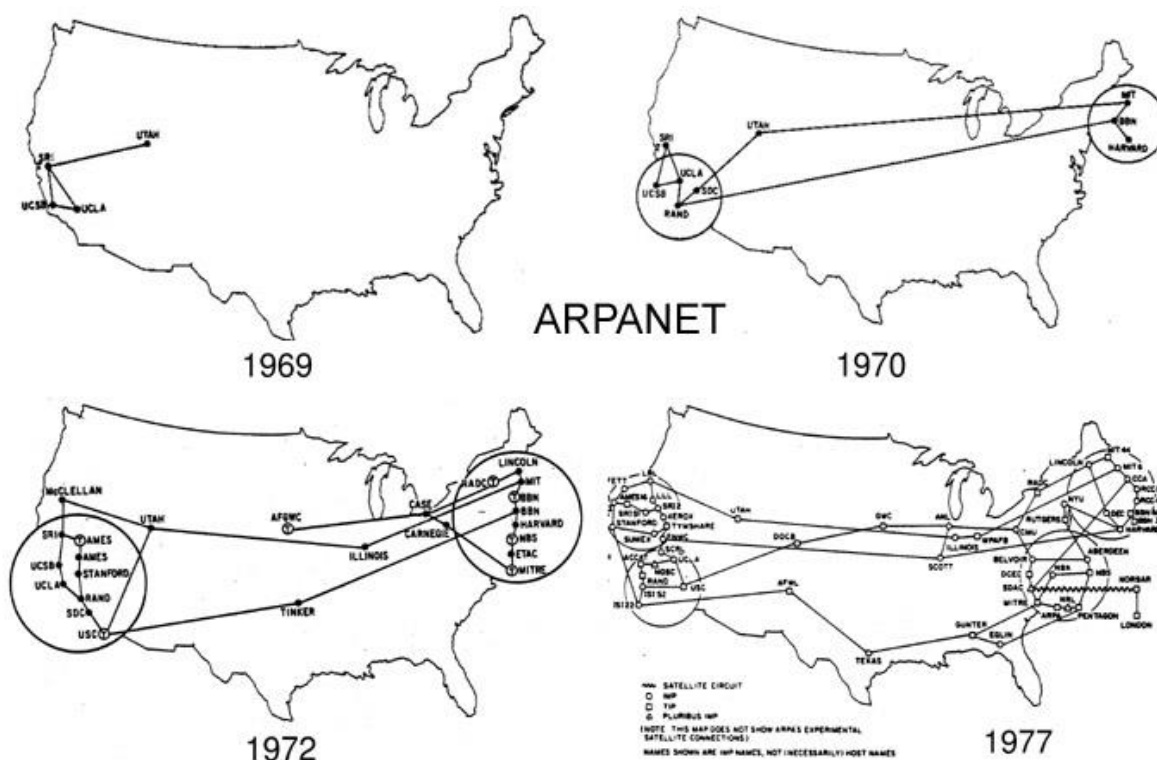
2.1.1 Arpanet

Jedny z nejstarších map internetu pocházejí z doby Arpanetu (1970-1990), ke kterému se postupně připojovaly další a další sítě a postupem času se celá síť přeměnila na Internet, jak jej známe dnes.

Na obrázku 7 je mapa sítě Arpanet a dalších připojených sítí z ledna 1983, v době kdy se ještě mapa všech připojených sítí vešla na jeden obrázek. Na obrázku 8 je ukázka rozšiřování sítě Arpanet o další sítě na území USA v letech 1969-1977.



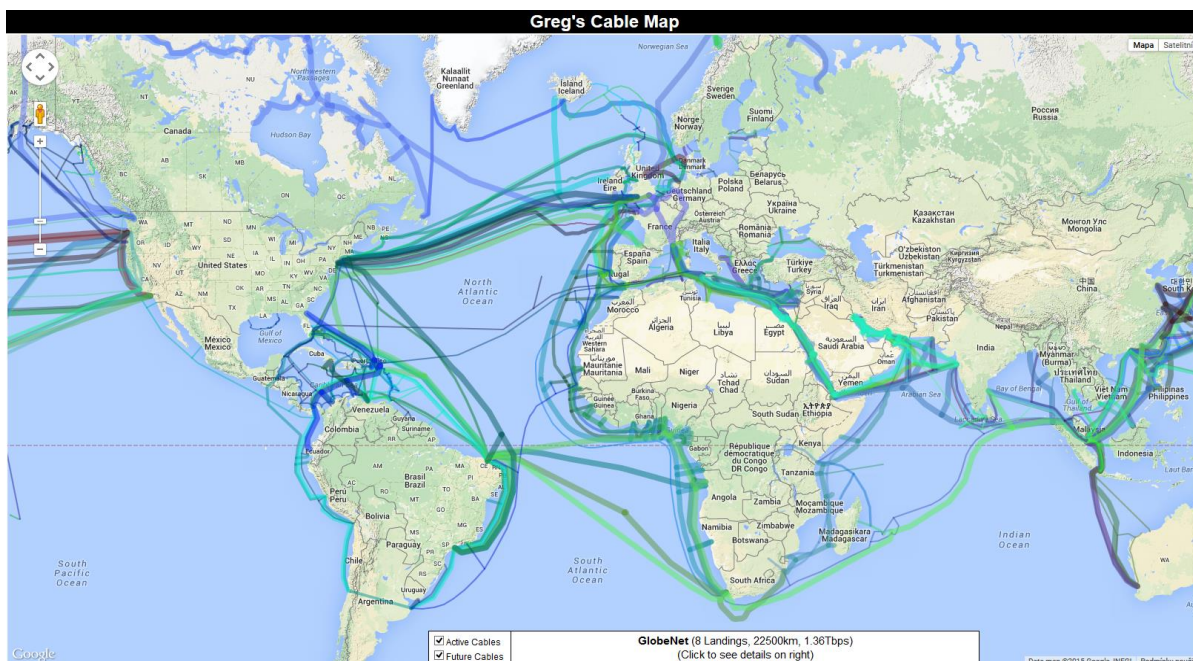
Obrázek 7 Síť Arpanet a sítě k ní připojené[17]



Obrázek 8 Rozšíření sítě v USA v letech 1969-1977[16]

2.1.2 Greg's Cable Map

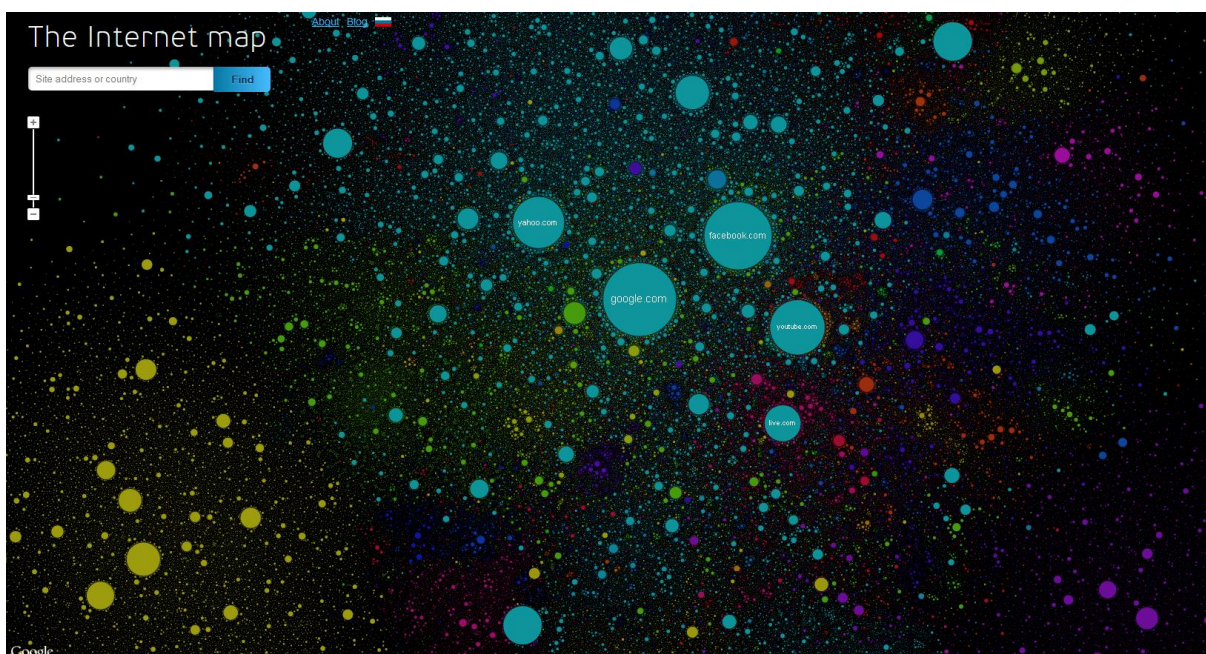
Mapa podmořských kabelů založená na datech z veřejně dostupných zdrojů, některé trasy jsou přesné, jiné převzaté pouze z marketingových materiálů. Data byla získávána zejména z Wikipedie a dále pomocí vyhledávání na Google.com. Zobrazuje aktuální i budoucí linky (ty které jsou ve výstavbě a jsou známy plány). Obsahuje pouze trasy nad 1Gbps, neboť jsou důležitější pro fungování Internetového spojení mezi zeměmi a menší linky (ve většině případů starší některé zrušené) by zbytečně zabíraly místo na mapě mapa by ztrácela na přehlednosti. Mapa je postavena na Google Maps. Na obrázku 9 je ukázka mapy podmořských kabelů, zaměřeno na Atlantický oceán a spojení mezi Evropou a USA kde vede nejvíc kabelů vedle sebe.



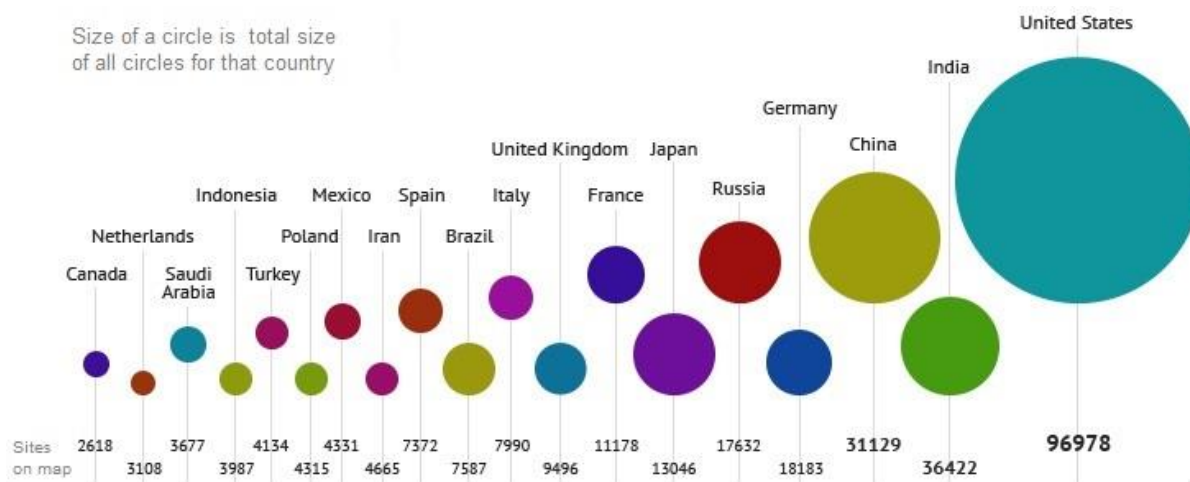
Obrázek 9 Mapa podmořských kabelů[9]

2.1.3 Internetová mapa provozu na webových stránkách

Mapa provozu na jednotlivých webových stránkách. Dvourozměrná reprezentace spojení mezi webovými stránkami na Internetu. Každá stránka je zobrazena jako kruh na mapě, velikost kruhu odpovídá provozu dané stránky. Na obrázku 10 je pohled na celkovou mapu, největší kruhy (i provoz) mají aktuálně stránky google.com, facebook.com, yahoo.com, youtube.com. Na obrázku 11 je míra provozu v jednotlivých zemích.



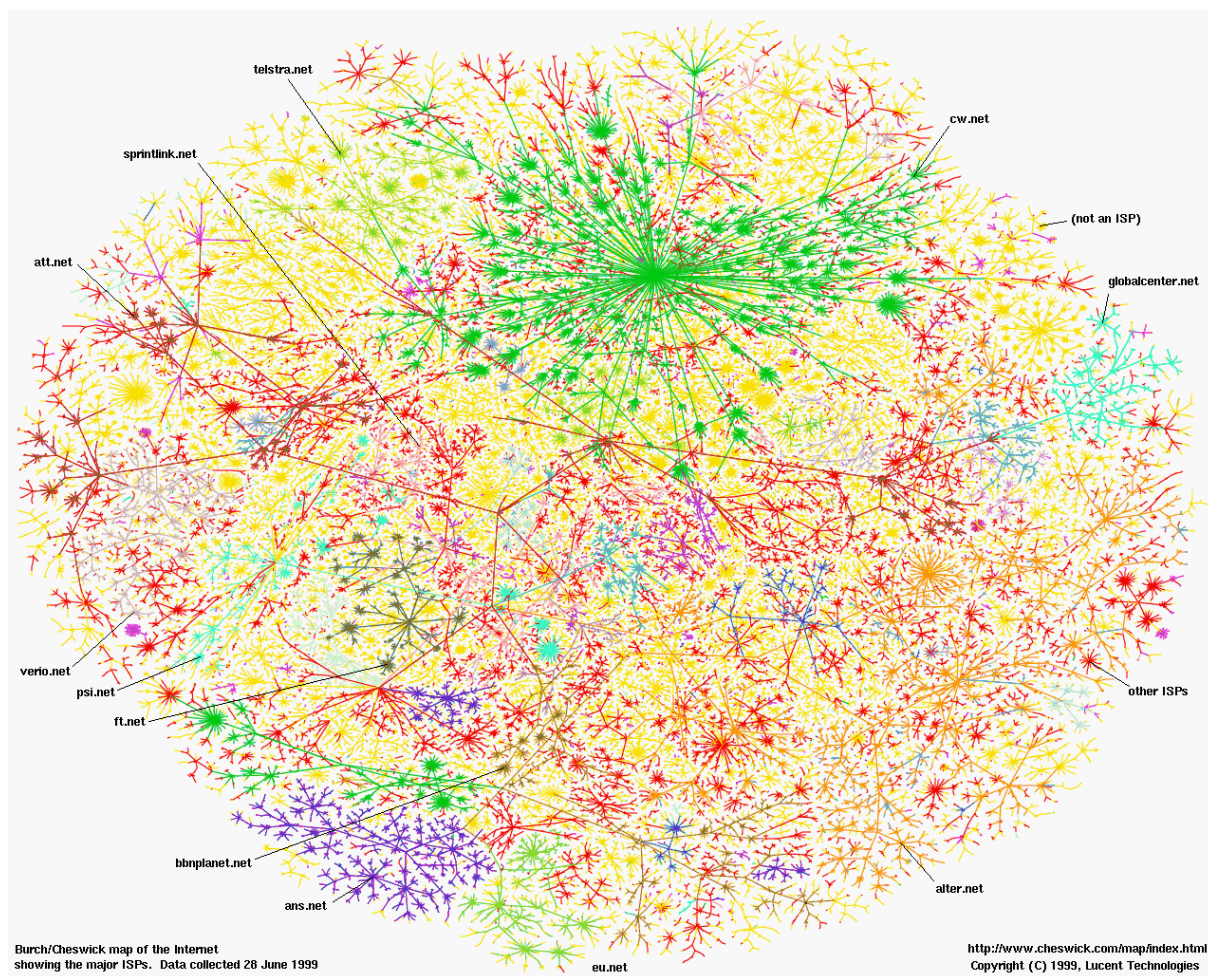
Obrázek 10 Pohled na celkovou mapu provozu[18]



Obrázek 11 Míra provozu v jednotlivých zemích[18]

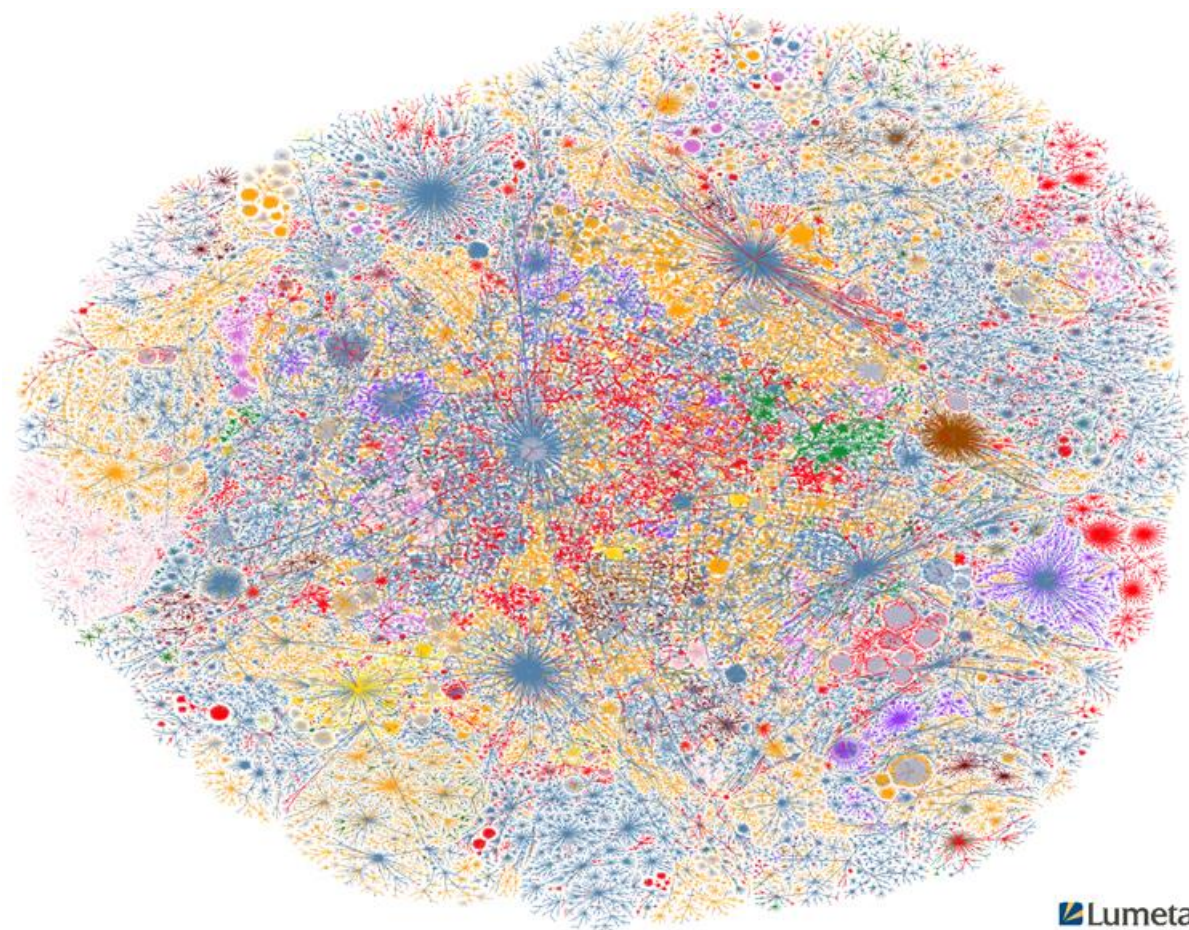
2.1.4 Internet Mapping Project

Projekt mapování začal William Cheswick a Hal Burch v létě 1998 v laboratořích Bell. Cílem bylo dlouhodobě získávat a ukládat topologická data. Tato data byla použita ke studiu směrovacích problémů, změn v síti, **DDos** (Distributed Denial of Service) útoků a teorie grafů. Získávání dat probíhalo pomocí trasování. V době předání projektu korporaci Lumeta obsahovala mapa asi 100 000 uzlů. Na obrázku 12 je mapa hlavních poskytovatelů z 28. června 1999.



Obrázek 12 Mapa hlavních poskytovatelů[10]

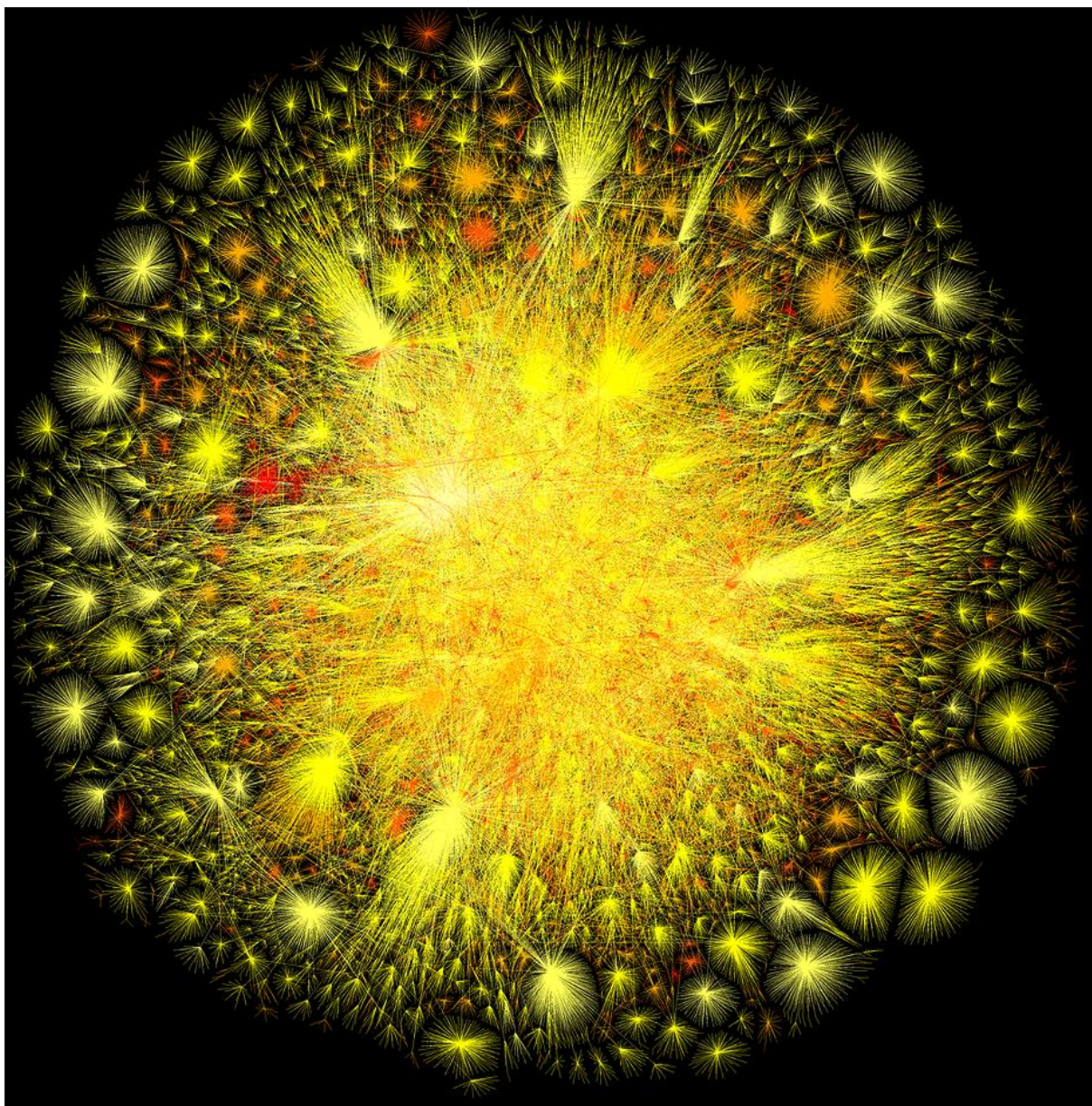
V roce 2000 byl projekt předán firmě Lumeta Corporation. V dnešní době firma Lumeta Corporation nabízí celou řadu komerčních řešení od průzkumu stavu sítě, bezpečnosti, dlouhodobého monitorování a dalších služeb spojených se správou sítě. Na obrázku 13 je mapa z 27. července 2014, na této mapě jsou pouze **IPv4** zařízení. Mimo mapy IPv4 vzniklé z původního projektu nabízí i mapu **IPv6** zařízení, která začala vznikat v roce 2005. Projekt mapování stále pokračuje a každý den jsou přidávány další záznamy.



Obrázek 13 Mapa firmy Lumeta[11]

2.1.5 The Opte Project

Projekt tvorby mapy sítě pro vizualizaci do podoby mapy nebo obrázku. Původní cíl projektu byl vytvořit obrázek nebo mapu sítě s využitím jediného počítače a jediného připojení k Internetu za 24 hodin. Projekt vytvořil Barrett Lyon v roce 2003-2004. V roce 2010 vznikl obrázek Internetu (Obrázek 14) publikovaný v časopise Discovery Magazine a v Muzeu Moderního Umění v New Yorku.



Obrázek 14 Mapa Internetu projektu OPTE[20]

2.2 Databáze IP adres

Každou **IP** adresu vlastní nějaká společnost, nebo osoba. V databázích vlastníků adres lze najít další informace, pro zpřesnění mapy sítě: adresa, jméno vlastníka, domény registrované na dané **IP** adrese, atd. Na mezinárodní úrovni spravuje přidělování **IP** adres korporace **ICANN**, která přiděluje adresy národním poskytovatelům.

2.2.1 ICANN

Celosvětový poskytovatel **IP** adres, stará se o správu **IP** adres, kořenových **DNS** serverů a nejvyšších domén (cz, com, eu, ...). Například v databázi **IPv4** adres (Obrázek 15) je uvedeno kdo vlastní nebo spravuje první část **IP** adresy, kdy byla přidělena a jaký je současný stav.

Prefix	Designation	Date	WHOIS	RDAP	Status [1]	Note
000/8	IANA - Local Identification	1981-09			RESERVED	[2]
001/8	APNIC	2010-01	whois.apnic.net		ALLOCATED	
002/8	RIPE NCC	2009-09	whois.ripe.net		ALLOCATED	
003/8	General Electric Company	1994-05	whois.arin.net		LEGACY	
004/8	Level 3 Communications, Inc.	1992-12	whois.arin.net		LEGACY	
005/8	RIPE NCC	2010-11	whois.ripe.net		ALLOCATED	
006/8	Army Information Systems Center	1994-02	whois.arin.net		LEGACY	
007/8	Administered by ARIN	1995-04	whois.arin.net		LEGACY	
008/8	Level 3 Communications, Inc.	1992-12	whois.arin.net		LEGACY	
009/8	IBM	1992-08	whois.arin.net		LEGACY	
010/8	IANA - Private Use	1995-06			RESERVED	[3]
011/8	DoD Intel Information Systems	1993-05	whois.arin.net		LEGACY	
012/8	AT&T Bell Laboratories	1995-06	whois.arin.net		LEGACY	
013/8	Administered by ARIN	1991-09	whois.arin.net		LEGACY	
014/8	APNIC	2010-04	whois.apnic.net		ALLOCATED	[4]
015/8	Hewlett-Packard Company	1994-07	whois.arin.net		LEGACY	
016/8	Digital Equipment Corporation	1994-11	whois.arin.net		LEGACY	
017/8	Apple Computer Inc.	1992-07	whois.arin.net		LEGACY	
018/8	MIT	1994-01	whois.arin.net		LEGACY	
019/8	Ford Motor Company	1995-05	whois.arin.net		LEGACY	
020/8	Computer Sciences Corporation	1994-10	whois.arin.net		LEGACY	
021/8	DDN-RVN	1991-07	whois.arin.net		LEGACY	
022/8	Defense Information Systems Agency	1993-05	whois.arin.net		LEGACY	
023/8	ARIN	2010-11	whois.arin.net		ALLOCATED	
024/8	ARIN	2001-05	whois.arin.net		ALLOCATED	
025/8	UK Ministry of Defence	1995-01	whois.ripe.net		LEGACY	
026/8	Defense Information Systems Agency	1995-05	whois.arin.net		LEGACY	
027/8	APNIC	2010-01	whois.apnic.net		ALLOCATED	
028/8	DSI-North	1992-07	whois.arin.net		LEGACY	
029/8	Defense Information Systems Agency	1991-07	whois.arin.net		LEGACY	
030/8	Defense Information Systems Agency	1991-07	whois.arin.net		LEGACY	
031/8	RIPE NCC	2010-05	whois.ripe.net		ALLOCATED	
032/8	Administered by ARIN	1994-06	whois.arin.net		LEGACY	
033/8	DLA Systems Automation Center	1991-01	whois.arin.net		LEGACY	
034/8	Halliburton Company	1993-03	whois.arin.net		LEGACY	
035/8	Administered by ARIN	1994-04	whois.arin.net		LEGACY	
036/8	APNIC	2010-10	whois.apnic.net		ALLOCATED	
037/8	RIPE NCC	2010-11	whois.ripe.net		ALLOCATED	
038/8	PSINet, Inc.	1994-09	whois.arin.net		LEGACY	
039/8	APNIC	2011-01	whois.apnic.net		ALLOCATED	
040/8	Administered by ARIN	1994-06	whois.arin.net		LEGACY	

Obrázek 15 Databáze IPv4 rozsahů adres[21]

2.2.2 Databáze IP adres www.czdomeny.cz

Databáze Českých domén, k vyhledané doméně nebo **IP** adrese zobrazí rozsah adres, majitele včetně adresy a další domény hostované na stejné **IP** adrese. Na obrázku 16 je reverzní záznam adresy pro **IP** adresu 77.75.72.3.

Reverzní záznam IP adresy 77.75.72.3

Reverzní záznam 77.75.72.3 = www.seznam.cz.

Informace o ip adrese 77.75.72.3:

77.75.72.0 - 77.75.72.255
descr: Seznam.cz
tech-c: SZN5-RIPE
source: RIPE # Filtered
org-name: Seznam.cz, a.s.
address: ing. Michal Feix
address: Prague 5
fax-no: +420234694115

netname: SEZNAM-CZ
country: CZ
status: ASSIGNED PA

org-type: LIR
address: Radlicka 3294/10
address: CZECH REPUBLIC

org: ORG-SA508-RIPE
admin-c: SZN5-RIPE
mnt-by: SEZNAM-MNT
organisation: ORG-SA508-RIPE
address: Seznam.cz, a.s.
address: 15000
phone: +420234694111

Další služby možné využít při doméne 77.75.72.3:

[[ping](#)] [[trace route](#)] [[http hlavička](#)]

Seznam .cz domén, které jsou hostovány na ip: 77.75.72.3

napoveda.cz

szn.cz

Obrázek 16 Reverzní záznam IP adresy[12]

2.2.3 Databáze IP adres <https://db-ip.com/>

Databáze více než 7 milionů záznamů, téměř polovina z USA. Obsahuje **IPv4** záznamy i některé **IPv6** záznamy (zhruba 7%). Tato databáze je částečně volně ke stažení, kompletní verze je placená.

Při vyhledávání zobrazí i polohu na mapě, pokud o ní existuje záznam. Na obrázku 17 je záznam informací o doméně www.microsoft.com, na mapě je i zobrazená poloha ve městě Cambridge v USA.

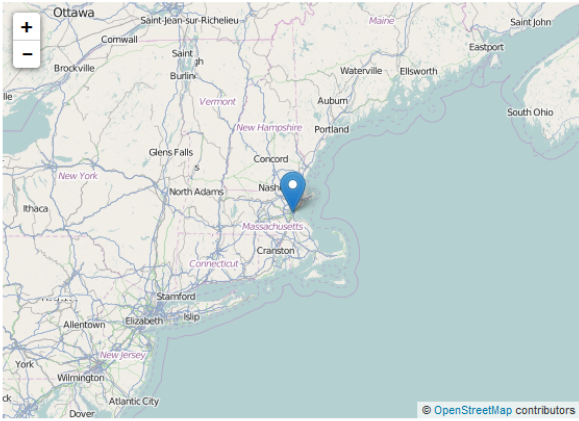
dbip

AboutDatabaseAPITools

IP address 172.227.70.234

Address type	IPv4
Hostname	a172-227-70-234.deploy.static.akamaitechnologies.com
ISP	Akamai Technologies
Organization	Akamai Technologies, Inc.
Timezone	America/New_York (UTC-4)
Local time	16:29:05
Country	United States
State / Region	Massachusetts
City	Cambridge
Coordinates	42.3636, -71.0825

Is the above data incorrect ? Help us improve our database accuracy.
[Report wrong data](#)



© OpenStreetMap contributors

View Larger Map

Copyright © 2015 db-ip.com, all rights reserved.
[terms of service](#) | [contact](#)

Obrázek 17 Záznam pro www.microsoft.com[19]

3. Síťová zařízení

Každé síťové rozhraní, které má **IP** adresu a může být dosaženo pomocí příkazů ping a traceroute lze pomocí těchto příkazů zmapovat a nezáleží, zda jde o router, server, nebo počítač či notebook.

3.1 Hub (opakovač)

Aktivní síťové zařízení pracující na fyzické vrstvě (1. vrstva). Umožňuje větvení sítě, prodlužuje maximální délku segmentu. Př.: 100 metrů u kabelu **cat5e** v Ethernetu. Po 100 metrech už signál začíná být zkreslený a mohou vznikat přeslechy z důvodu rušení signály jiných zařízení a kabelových vedení. Opakovač se vůbec nestará o data, která do něj přicházejí, pouze přijatý signál v podobě 1 a 0 obnoví a odešle na všechny porty. Zpoždění opakovače je tak pouze 1 bit. Opakovače jsou v dnešní době nahrazovány přepínači, které jsou lepší z hlediska bezpečnosti. Neodesílají data na všechny porty ale pouze správným směrem. Na síti není vidět z pohledu **ICMP** paketů, nemá **IP** adresu, neprovádí kontrolu chyb, ani žádná rozhodnutí (přepínání a směrování), a nemění data procházející skrz opakovač.

3.2 Switch a Bridge (Přepínač a most – dvouportový přepínač)

Aktivní síťové zařízení pracující na linkové vrstvě (2. vrstva). Odděluje od sebe segmenty jedné sítě. Odděluje kolizní doménu, protože nepřeposílá rámce na všechny porty, ale pouze na jeden, uživatel sítě tak nemůže vidět komunikaci v jiném segmentu sítě.

Způsoby preposílání rámců

Store and forward (ulož a pošli) – celý rámec je nejprve přijat uložen do paměti, zkontrolován kontrolní součet a teprve pak je odeslán na příslušný port. Nejpomalejší metoda vhodná do sítí s velkým množstvím kolizí.

Cut through nebo **On the fly** (za letu) – rámec je odeslán na příslušný port hned jak je to možné (po načtení cílové MAC adresy). Snižuje nároky na paměť, zvyšuje rychlost přepínání, ale může odeslat i poškozený paket.

Fragment free – rámec se začne preposílat po přijetí 64 bytů, kdy je jisté, že nevznikla kolize během odesílání.

Adaptive switching – přepínání mezi metodami Store and forward a Cut through.

Dokonalejší (a dražší) verze přepínačů, na rozdíl od běžných přepínačů umí směřovat pakety podobně jako směrovače

3.3 Router (Směrovač)

Aktivní síťové zařízení pracující na síťové vrstvě (3. vrstva). Spojuje a odděluje od sebe dvě nebo více sítí (v každé síti je jiný rozsah **IP** adres). Směrovač může být specializované zařízení nebo obyčejný počítač. Specializované směrovače však nabízí vyšší rychlost. Ke směrování se využívá směrovací tabulka se záznamy nejlepší cesty sítí k cíli. Směrovač může fungovat jako brána do internetu, kdy na jedné straně je vnitřní síť a na druhé je připojen k internetu, uvnitř sítě se používají privátní adresy, které jsou ve směrovači překládány na adresu tohoto směrovače, aby mohly pakety putovat sítí.

Některé směrovače po zahození datagramu negenerují zprávu **ICMP** time exceeded, ale při dostatečné hodnotě **TTL** předají datagram dále. Odpověď se neposílá nejspíš z důvodu bezpečnosti, aby směrovač neprozradil svoji IP adresu.

4. Java technologie

Programovací jazyk Java nabízí nejrůznější specializované knihovny, v této kapitole bude popsána knihovna **Jpcap**, kterou vytvořil a upravoval K. Fujii[5]. Dále zde budou popsány technologie pro tvorbu grafických rozhraní.

4.1 Knihovny WinPcap a Jpcap

Knihovna WinPcap obsahuje nástroje pro přístup na linkovou vrstvu ve Windows, umožňuje zachytávání a odesílání paketů. Knihovna **Jpcap** využívá tuto knihovnu ve Windows, v OS Linux není potřeba.

Jpcap je externí knihovna, která slouží pro zachytávání síťových paketů v jazyce Java. Ve Windows využívá knihovnu WinPcap.

4.1.1 Třídy knihovny Jpcap

Třída jpcap.Packet - Paket z knihovny Jpcap obsahuje:

int caplen a int len – integer obsahující délku zachyceného paketu v bytech

byte[] header – bytové pole obsahující kombinaci hlavičky Ethernet, IP hlavičky a dalších (TCP, UDP, ICMP atd.)

byte[] data – bytové pole obsahující data

DatalinkPacket datalink – rámec linkové vrstvy, umožňuje vypsát zdrojovou, cílovou MAC adresu a typ rámce (tyto data jsou uloženy na začátku pole header)

long sec a long usec – čas zachycení datagramu v unixovém tvaru (př.: 1363011466139) na datum se převede pomocí konstruktoru

Date(packet.sec*1000 + packet.usec/1000)

Třída Jpcap.NetworkInterface - Třída reprezentující síťové rozhraní. Před zahájením zachytávání nebo odesílání se musí vybrat rozhraní, které bude použito.

Třída Jpcap.JpcapCaptor - Třída umožňující zachytávání paketů na vybraném síťovém rozhraní, a čtení paketů ze souboru.

Třída Jpcap.Jpcap Sender - Třída pro odesílání vlastních vytvořených paketů.

Třída Jpcap.Jpcap Writer - Třída pro zápis paketů do souboru.

4.2 Java API pro tvorbu GUI

AWT - Abstract Window Toolkit - vydána v roce 1995 společně s JDK 1.0. Sada tříd a metod určená pro tvorbu grafických rozhraní v jazyce Java. Obsahuje chyby v návrhu, zejména způsobené závislostí na platformě, což odporuje základním vlastnostem jazyka Java.

Swing - vydán v roce 1997. Nabízí sadu „lehkých“ komponent pro tvorbu grafických rozhraní. Opravuje chyby v návrhu, ale stále založeno na rodičích z AWT.

4.2.1 JavaFX

Představena v roce 2007 (původní vývoj Chris Oliver ve firmě SeeBeyond). Vyvíjena s cílem nahradit Swing pro tvorbu grafických rozhraní. Firmu SeeBeyond převzal Sun a následně v roce 2009 převzal firmu Sun Oracle. JavaFX je vhodná pro tvorbu Rich web aplikací i GUI aplikací na různých platformách. Podporuje dotyková zařízení, zachytává eventy onTouch, onSwipe, onRotate. Dále má rozsáhlou podporu pro tvorbu grafů. Může MVC model, rozmístění tlačítek a dalších prvků v okně je odděleno od ovládání a vlastního kódu programu. Model je obsažen v datových strukturách a kódu programu, View je v souboru FXML a Controller je defaultně v souboru FXMLDocumentController.java.

Velký rozdíl je například v kreslení, v grafickém rozhraní Swing se nakreslí úsečka *drawLine(x1 ,y1 ,x2, y2)* a při posunu jednoho koncového bodu tažením myši se musí úsečka smazat a vytvořit nová, například pomocí XOR kreslení. V rozhraní JavaFX je úsečka jako datová struktura `Line l = new Line(x1, y2, x2, y2);` Pro posun jednoho konce stačí volat metody `l.setStartX(x);` a `l.setStartY(y);` nebo `l.setEndX(x);` a `l.setEndY(y);`. Navíc při přibližování se tato úsečka chová jako vektorová grafika (je definován první a poslední bod) a neztrácí kvalitu.

Praktická část

5. Návrh aplikace

Pro získávání dat bylo třeba implementovat zachytávání a odesílání paketů po síti. Pro ukládání dat byla vytvořena databáze. Dále bylo třeba navrhnout vhodné zobrazování vytvořené mapy na obrazovku.

Funkční požadavky

Cílem praktické části bylo vytvořit aplikaci na odesílání a zachytávání paketů **ICMP**.

A tvorbu mapy internetových zařízení

Funkční požadavky dle zadání byly následující:

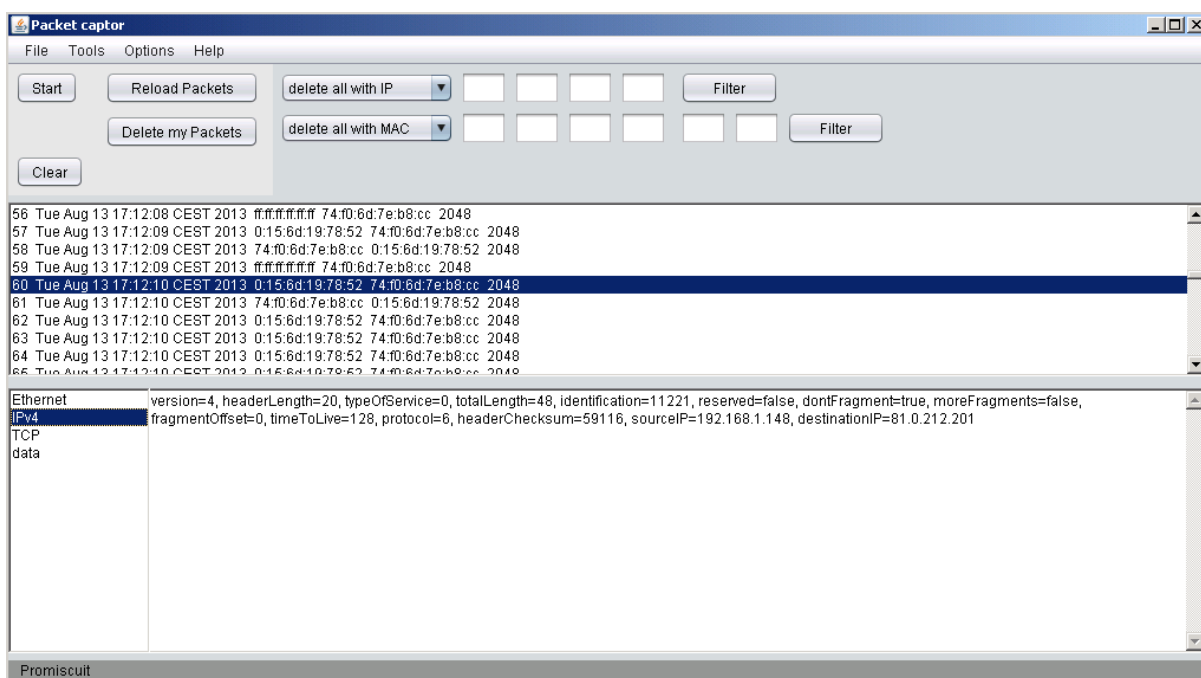
1. Zachytávání a odesílání paketů na síťovém rozhraní
2. Ukládání a načítání uzlů sítě do databáze
3. Vykreslování na obrazovku

Nefunkční požadavky

Aplikace byla vyvíjena v operačním systému Microsoft Windows 8.1 64bit a Microsoft Windows XP 32bit s rozšířením knihovnamí Jpcap a WinPcap. Pro funkčnost v 64bitovém systému je třeba nainstalovat 32bitovou verzi Java Development Kit. Grafické uživatelské rozhraní je určeno pro rozlišení 1024x600 a vyšší.

5.1 Základ aplikace

Základem aplikace se stala má bakalářská práce Zachytávání a analýza síťových paketů v jazyce Java (Obrázek 18). Další část pro spojení s databází tvoří semestrální práce z předmětu INPDA – programování databázových aplikací, vyvíjená jako prostředek pro ukládání dat z diplomové práce do databáze.



Obrázek 18 Okno aplikace pro zachytávání a analýzu paketů

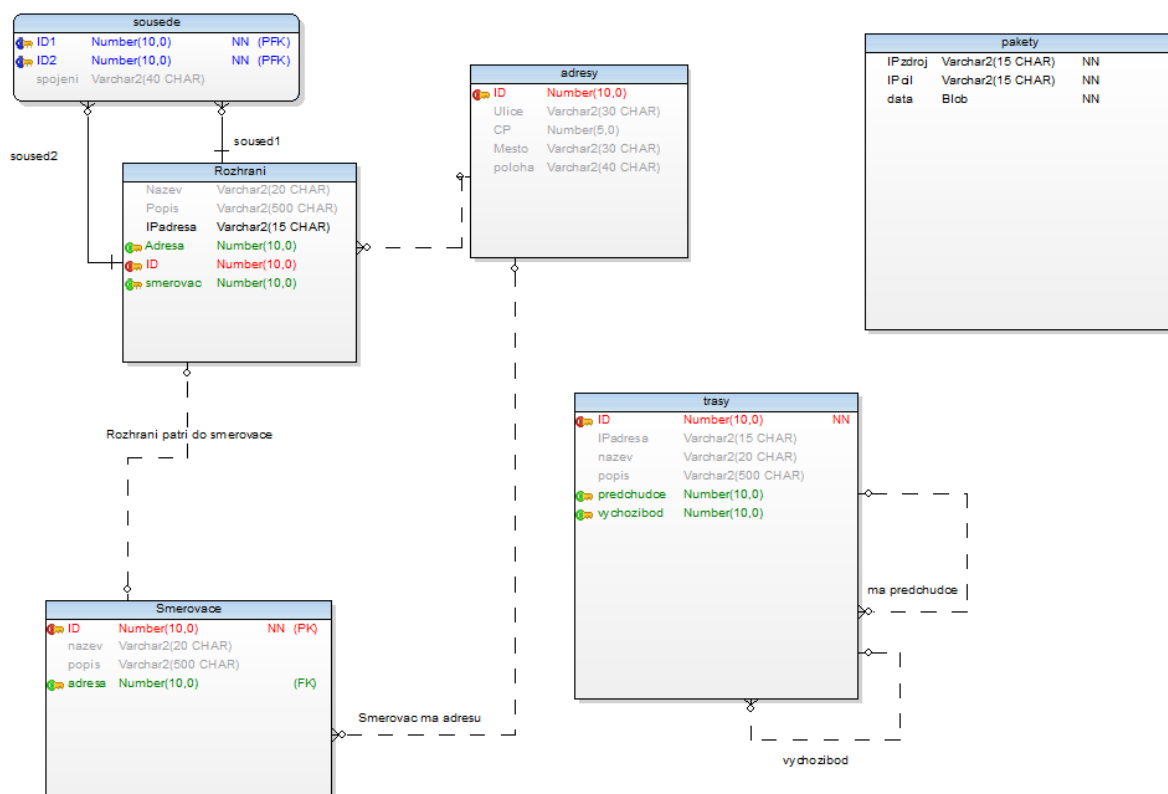
5.2 Návrh databáze

Databáze slouží k ukládání zpracovaných dat (paketů), nezpracované pakety budou ukládány do souboru, protože knihovna Jpcap podporuje ukládání a načítání ze souboru. A ukládání velkého množství paketů do databáze by bylo zbytečné. Tabulka kam by se ukládaly zachycené pakety by obsahovala sloupec typu **BLOB** s paketem a možná pár informativních sloupců o paketu (**IP** adresy, protokoly, atd), které by byly duplicitně v paketů i v dalších sloupcích, v návrhu je zobrazena vpravo bez relací k ostatním tabulkám. Navíc při zachycené komunikaci by tabulka obsahovala řádově tisíce paketů odeslaných z počítače kde běží aplikace k cíli.

5.2.1 Model databáze

Původní model (Obrázek 19) obsahuje 6 tabulek, z nichž jedna představuje M:N relaci. Tabulka trasy je připravena pro záznam tras příkazu tracer (počítá se s opakováním záznamů), v tabulce rozhraní bude každé rozhraní pouze jednou a v tabulce sousedé budou sousední rozhraní. Jednotlivá rozhraní budou sdružována do směrovačů, pokud bude nalezen způsob jak zjistit, které rozhraní patří ke směrovači. Na obrázku 20 je aplikace pro ovládání této databáze.

Protože databáze bude sloužit pro ukládání dat pro diplomovou práci, bude mít pouze jednoho uživatele s jednou rolí k přístupu k datům.



Obrázek 19 Původní návrh databáze vytvořený v rámci semestrální práce

Databaseová aplikace

Načti rozhraní

Načti směrovače

Načti adresy

Načti trasy

Vyčisti tabulku

Nový záznam

Odstranit záznam...

Master detail

Hledat IP adresu

Hledat

Ukončit

ID	NAZEV	POPIS	IPADRESA	ADRESA	SMEROVAC
1					
2	doma	AP na střeše	192.168.1.100	1	
8	Sila	Sila na rakety zrní-zrní u Hrubešů	10.11.2.254	1	
9	Správce		10.11.2.149	1	
11	NET.UPC	CZ-PRA-POP1-RB1.NET.UPC.CZ	213.192.6.21		
12	aorta.net		84.116.131.153		
13	nix5.seznam.cz		91.210.16.195		
14			77.75.75.222		
15	www.seznam.cz		77.75.72.3		
17	nix2.seznam.cz		91.210.16.194		
18			77.75.75.210		
19	www.seznam.cz		77.75.76.3		
20	rozhraní jedna	Do internetu	1.1.1.1	17	0
21	rozhraní dva	K hradu Duloc	2.2.2.2	17	0
22	rozhraní tři	K perníkové chaloupce	3.3.3.3	17	0
23	rozhraní čtyři	K chaloupce sedmi trpaslíků	4.4.4.4	17	0
24	nix3.cesnet.cz	nix3-20ge.cesnet.cz	91.210.16.191		
25			105.143.131.105		
11	NET.UPC	CZ-PRA-POP1-RB1.NET.UPC.CZ	213.192.6.21		
13	nix5.seznam.cz		91.210.16.195		
17	nix2.seznam.cz		91.210.16.194		
24	nix3.cesnet.cz	nix3-20ge.cesnet.cz	91.210.16.191		

Obrázek 20 Ukázka okna aplikace pro ovládání databáze

6. Vývoj aplikace

V průběhu tvorby aplikace docházelo ke změnám databáze, aby vyhovovala potřebám aplikace. Dále k implementaci technologií pro práci se sítí a k tvorbě datových struktur pro práci s uzly a pro zobrazování.

Zachytávání paketů bylo vyřešeno v bakalářské práci, zbývalo vyřešit vytvoření paketu a jeho odeslání na určitou adresu, což řeší část knihovny JPCAP.Sender. Dále přijetí odpovědi a její zpracování.

6.1 Úpravy databáze

Analyzovaná data jsou ukládána do databáze, která byla vytvořena v rámci předmětu INPDA a upravena podle nově vzniklých požadavků přidáním sloupců do tabulky *rozhrani* pro vykreslování na obrazovku pro virtuální mapu a pro mapu s reálným podkladem, dále odebráním zbytečných tabulek *pakety* a *trasy*.

6.1.1 Odebrání tabulky trasy

Tabulka *trasy* měla původně sloužit k ukládání nově zjištěných tras, před jejich přesunem do tabulky *rozhrani*, nově jsou získané trasy ukládány přímo do tabulky *rozhrani*.

6.1.2 Změna tabulky sousedé na předchůdci

Tabulka *sousedé* reprezentující M:N relaci mezi rozhraními obsahovala každé spojení 2x, což bylo zbytečné, proto bylo jedno spojení odebráno a tabulka přejmenována na *predchudci*. Spojení, které bylo ponecháno, reprezentuje spojení směrem ke zdroji, který provádí mapování. Dále bylo do tabulky *predchudci* přidán sloupec *vychozibod*, reprezentující zdroj (předpokládá se tvorba mapy z různých míst v síti).

6.1.3 Přidání sloupců pro polohu na mapě

Pro zobrazení rozhraní na mapě byly přidány sloupce *X* a *Y* pro virtuální mapu a *XMAPA* a *YMAPA* pro reálnou mapu.

6.1.4 Odebrání tabulky pakety

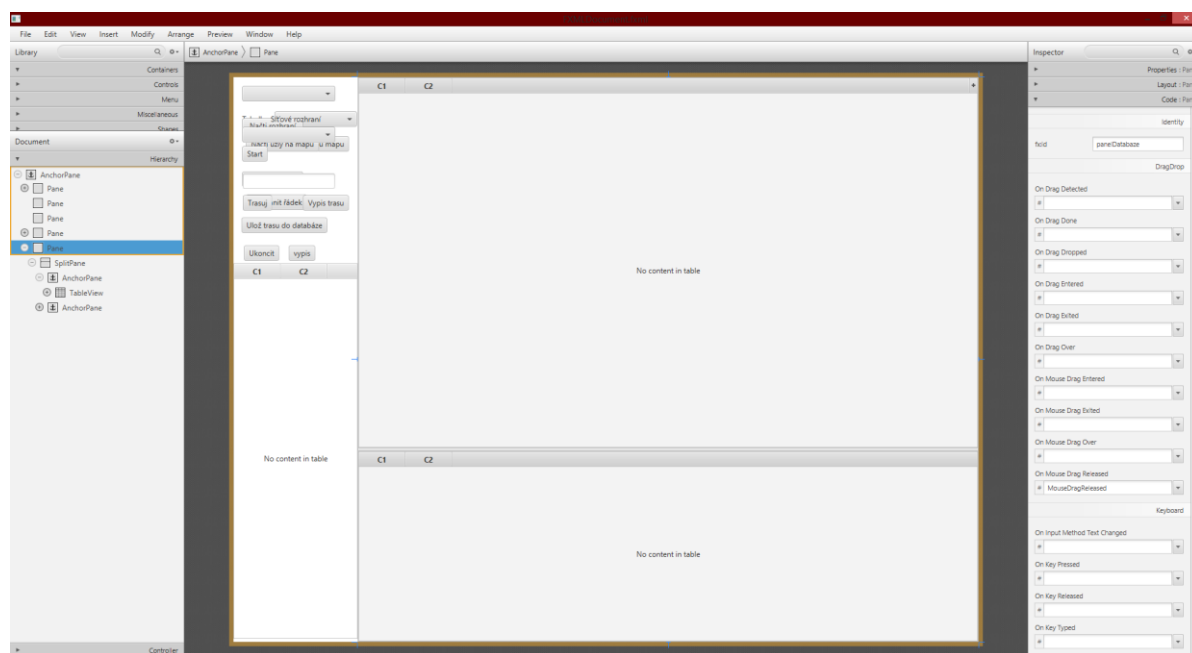
Protože zachycené pakety se ukládají do souboru pomocí knihovních funkcí knihovny JPCAP, byla tabulka na ukládání paketů do databáze zbytečná a byla odebrána.

6.2 Grafické rozhraní

Grafické rozhraní je vytvořeno pomocí **api** JavaFX. Rozložení prvků na obrazovce je podobné jako v semestrální práci pro přístup k databázi, ovládací panel je umístěn nalevo a zbytek obrazovky zabírá zobrazovací část.

6.2.1 Podpora v Netbeans

V prostředí Netbeans 8.0 není tak dokonalá podpora pro návrh a tvorbu grafických rozhraní JavaFX jako pro Swing a AWT (přepnutí do Design části a výběr z palety), ale existuje plugin Scene builder, který po otevření nabízí podobné funkce v novém okně, nicméně přepínání mezi okny Netbeans do Scene builderu a zpět není zrovna pohodlné, protože po zavření se musí Scene builder spouštět znovu, nebo nechat běžet Scene builder na pozadí a přepínat mezi okny. Rovněž je možná přímá editace souboru FXML. Na obrázku 21 je ukázka návrhu aplikace v pomoci aplikace JavaFX Scene Builder.



Obrázek 21 Ukázka návrhu aplikace v JavaFX Scene Builder

7. Struktura aplikace

Aplikace byla vyvíjena v jazyce Java SE 8, ve vývojovém prostředí NetBeans IDE 8.0.1 v operačních systémech Windows XP 32bit a Windows 8 64bit s rozšířením knihovnamí Jpcap a WinPcap. Pro vývoj v 64bitovém prostředí bylo třeba nainstalovat 32bitovou verzi Java Development Kitu. Databáze pro ukládání dat je Oracle 10g express edition.

7.0.1 Instalace vývojového prostředí

Jako první je třeba nainstalovat Java Development Kit. Po něm přichází na řadu vývojové prostředí NetBeans IDE 7.1.1. Po instalaci vývojového prostředí je třeba ještě nainstalovat pomocné knihovny Jpcap a WinPcap, obě knihovny mají klasický instalátor, který nainstaluje vše automaticky po odsouhlasení licenčního ujednání. Databáze Oracle 10g XE je instalována samostatně.

Pro samotné spuštění aplikace je třeba mít nainstalovanou podporu jazyka Java, a dále knihovny Jpcap a WinPcap a databázi.

7.1 Struktura Aplikace

Struktura aplikace je na obrázku 22. Spustitelná část aplikace je ve třídě *Aplikace.java*, kromě metody *main* obsahuje pouze metodu *start* pro spuštění okna popsaneho v souboru *FXMLDocument.fxml*. Soubor *FXMLDocument.fxml* obsahuje popis a rozmístění prvků v okně pomocí jazyka FXML (jazyk XML upravený pro popis prvků JavaFX), tím se odděluje grafická část view od ovládací části controller modelu MVC. Třída *FXMLDocumentController.java* obsahuje metody pro ovládání aplikace a pro volání metod z dalších tříd. Třída *OvladaniDB.java* obsahuje metody pro připojení k databázi, načítání tabulek a editaci prvků databáze. Třída *Uzel.java* dědí *javafx.scene.shape.Circle* pro vykreslení uzlu na obrazovku a dále obsahuje atributy a metody pro nastavení obsahu dle databáze (1 řádek z databáze se načte do 1 instance třídy *Uzel*).

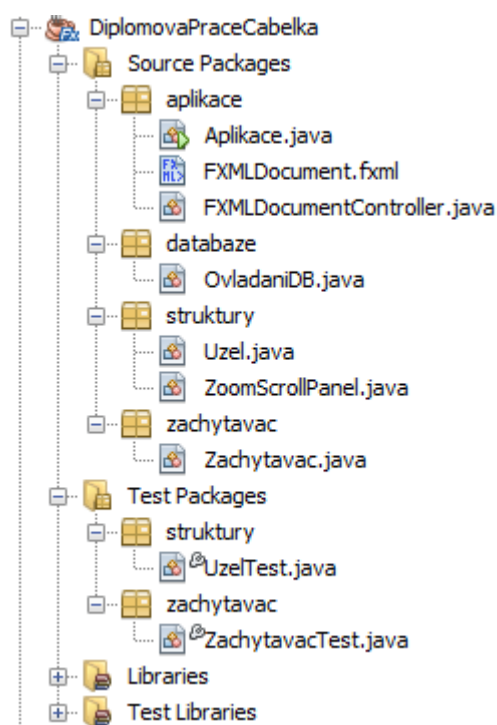
Třída *ZoomScrollPane.java* dědí *javafx.scene.control.ScrollPane*, definuje panel, který je možno zvětšovat, zmenšovat a pomocí metody *zoom(double zoom)* tyto transformace převede na veškerý obsah na tomto panelu (prvky jsou definovány podobně jako ve vektorové grafice, takže ani velké přiblížení nesnižuje kvalitu zobrazení).

Třída *Zachytavac.java* dědí *java.util.Observable*, obsahuje metody a atributy pro spojení aplikace s knihovnou *JPCAP.Captor*, v rámci této třídy se vytváří nové vlákno, které

je „pozorováno“ pomocí implementace návrhového vzoru Observer, hlavní část aplikace čeká na upozornění po registraci `zachytavac.addObserver(this)` po zachycení paketu je volána metoda `notifyObservers(packet)`; která upozorní všechny objekty, které sledují její stav. Po přijetí upozornění je ve sledující třídě spuštěna metoda `update`, která přijme a zpracuje nový paket, předaný jako argument.

Třída `OvladaniDB.java` slouží pro spojení s databází a obsahuje metody pro připojení, ukládání a načítání informací z databáze.

Dále zde jsou přítomny testovací třídy pro otestování některých tříd a metod.



Obrázek 22 Třídy projektu diplomové práce

7.2 Databáze

Finální verze databáze (Obrázek 23) obsahuje 4 tabulky: *rozhrani*, *predchudci*, *adresy* a *smerovace*. Pro vstup do databáze existuje jediný uživatel, protože databáze slouží k ukládání dat pouze z této aplikace.

Tabulky *rozhrani* a *predchudci*

Tabulka *rozhrani* reprezentuje síťový uzel (jedno síťové rozhraní směrovače, serveru, počítače). Tabulka *predchudci* představuje M:N relaci mezi jedním uzlem a jeho předchůdcem (rozhraní směrovače blíže ke zdroji trasování). Sloupce *X* a *Y* slouží pro

vykreslování na virtuální mapu, sloupce *XMAPA* a *YMAPA* slouží pro vykreslování na mapu s reálným podkladem. Každý uzel je tak v databázi pouze jednou, ale může mít různou pozici na dvou různých mapách. V tabulce 2 je význam a využití jednotlivých sloupců tabulky *rozhrani* a v tabulce 3 jsou sloupce z tabulky *predchudci*.

Tabulka 2 Význam sloupců tabulky *rozhrani*

ID	Number(10, 0) PK	Primární klíč, ID rozhraní
NAZEV	Varchar2(20 CHAR)	Název rozhraní, získaný například DNS překladem
IPADRESA	Varchar2(15 CHAR)	IPv4 nebo IPv6 adresa
POPIS	Varchar2(500 CHAR)	Doplňující informace
ADRESA	Number(10, 0) FK	Adresa rozhraní
SMEROVAC	Number(10, 0) FK	Směrovač do kterého patří rozhraní
X	Number(4, 0)	Pozice na virtuální mapě pro vykreslení
Y	Number(4, 0)	Pozice na virtuální mapě pro vykreslení
XMAPA	Number(4, 0)	Pozice na reálné mapě pro vykreslení
YMAPA	Number(4, 0)	Pozice na reálné mapě pro vykreslení

Tabulka 3 Význam sloupců tabulky *predchudci*

ID	Number(10, 0) PK	Primární klíč, ID rozhraní
IDPREDCHUDCE	Number(10, 0) PK	Primární klíč, ID předchůdce
VYCHOZIBOD	Number(10, 0)	Zdroj trasování (počítač kde běží aplikace)
SPOJENI	Varchar2(40 CHAR)	Informace o typu spojení (Wi-Fi, ethernet, ...)

Tabulka *adresy*

Tabulka *adresy* slouží pro uložení fyzické (reálné adresy, ne MAC) adresy rozhraní a GPS souřadnic.

Tabulka 4 Význam sloupců tabulky *adresy*

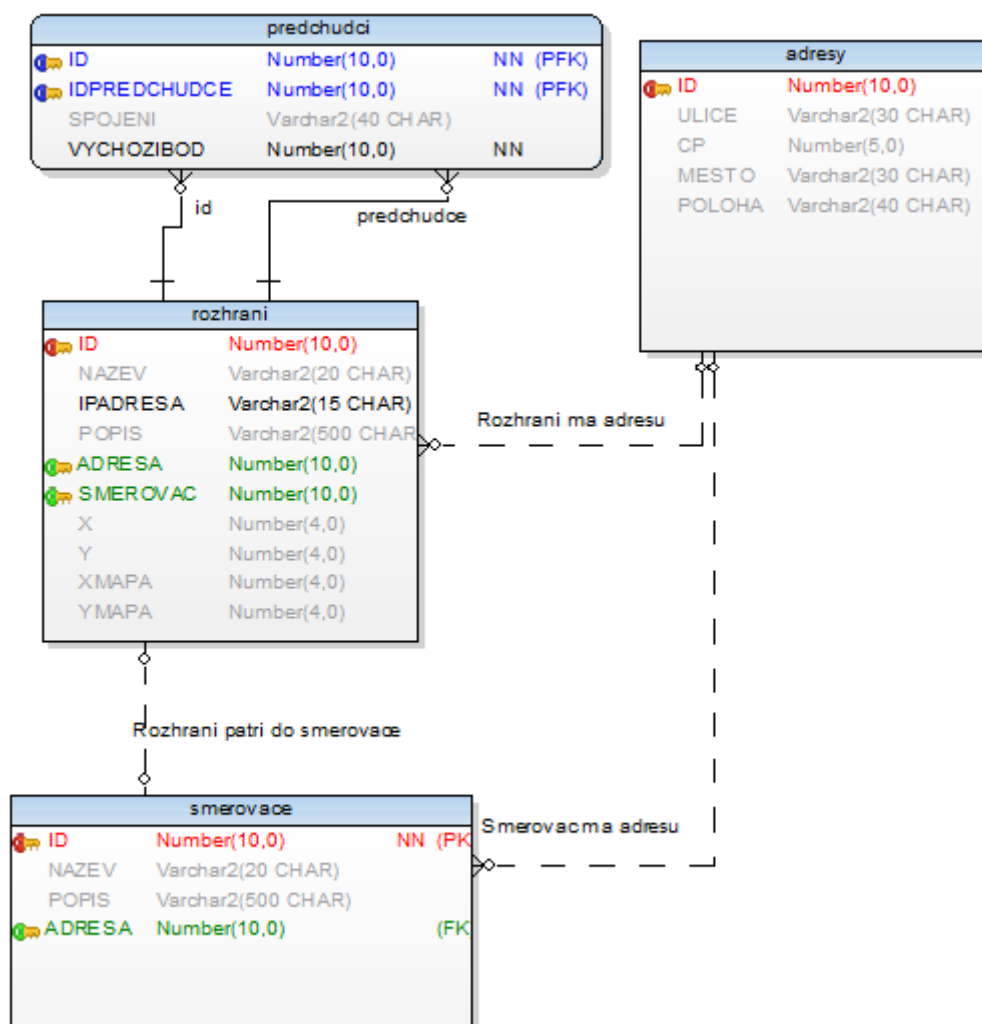
ID	Number(10, 0) PK	Primární klíč, ID adresy
ULICE	Varchar2(30 CHAR)	Ulice
CP	Varchar2(5 CHAR)	Číslo popisné
MESTO	Varchar2(30 CHAR)	Město
POLOHA	Varchar2(40 CHAR)	GPS poloha adresy

Tabulka *smervace*

Tabulka *smervace* slouží pro uložení směrovačů, které se skládají z několika rozhraní, nicméně při trasování z jednoho bodu lze vidět pouze rozhraní, které je nejbližší ke zdroji. Při trasování z několika bodů (umístěných v různých místech sítě) není poznat, která rozhraní patří ke stejnému směrovači, Proto tabulka není přímo využívána.

Tabulka 5 Význam sloupců tabulky *smervace*

ID	Number(10, 0) PK	Primární klíč, ID směrovače
NAZEV	Varchar2(20 CHAR)	Název směrovače
POPIS	Varchar2(500 CHAR)	Doplňující informace (info o majiteli, výrobce zařízení, ...)
ADRESA	Number(10, 0) FK	Adresa smerovače



Obrázek 23 Finální model databáze

7.3. Vstupy aplikace

Vstupem aplikace je vytvořená trasa od zdroje k vybrané cílové adrese pomocí paketů **ICMP**. Dále zachytávání na vybraném síťovém zařízení pomocí knihovny Jpcap pro zjištění provozu na síti a z něj další **IP** adresy. A načtení souboru s pakety uloženými pomocí knihovny Jpcap.

7.3.1 Tvorba trasy

První krok tvorby trasy je získání výchozí brány pro odeslání paketu, poté se s použitím **IP** adresy vytvoří **ICMP** datagram (formálně správně je datagram **ICMP**, obecně se používá název paket, z anglického packet) s hodnotou **TTL** = 1, je vložen do rámce ethernetu a odeslán přes zvolené síťové rozhraní na výchozí bránu.

Pro identifikaci paketu se používá pole *identifikace* z hlavičky **IPv4** s hodnotou **123654** a vlastní data v **ICMP** paketu „*datayg*“.

```
public ICMPPacket vytvorPaket(InetAddress srcIP, InetAddress dstIP,
    byte[] src_mac, byte[] dst_mac) {
    ICMPPacket icmp = new ICMPPacket(); // ICMP paket
    icmp.type = ICMPPacket.ICMP_ECHO;
    icmp.seq = 100;
    icmp.id = 0;
    icmp.hop_limit = 1;
    int idenfikace = 123654; // identifikace paketu
    icmp.setIPv4Parameter(0, false, false, false, 0, false, false, false,
        0, idenfikace, 1, ICMPPacket.IPPROTO_ICMP, srcIP, dstIP);
    icmp.data = "datayg".getBytes(); // uložení vlastních dat pro
    // identifikaci paketu
    EthernetPacket ether = new EthernetPacket(); // rámec ethernetu
    ether.frameType = EthernetPacket.ETHERTYPE_IP;
    ether.src_mac = src_mac; // zdrojová MAC adresa
    ether.dst_mac = dst_mac; // cílová MAC adresa (nejbližší switch/router,...)
    icmp.dataLink = ether;
    return icmp;
}
```

```

sender.sendPacket(icmpTracert);
while (true) {
    ICMPPacket p = (ICMPPacket) captor.getPacket();
    pakety.add(p);
    if (p == null) {
        txtVypis.setText("chyba zacina znovu");
        HledejTrasu();
    } else if (((p.data[3] & 0xFF) == 34) && ((p.data[4] & 0xFF) == 227)) {
        if (p.type == ICMPPacket.ICMP_TIMXCEED) { //uspech, na ceste
            txtVypis.setText("na cestě");
            trasa.add(p);
            lvPakety.getItems().add(p.toString());
            icmpTracert.hop_limit++; //zvyseni TTL o 1
        } else if (p.type == ICMPPacket.ICMP_UNREACH) {
            //cil nedostupný, konec trasování
            txtVypis.setText("cil nedostupný, konec");
            trasa.add(p);
            lvPakety.getItems().add(p.toString());
            break;
        }
        sender.sendPacket(icmpTracert);
    } else if (p.data.length == 6) {
        //kontrola dat vlozenych do paketu "datayg"
        if (((p.data[0] & 0xFF) == 100) && ((p.data[1] & 0xFF) == 97)
            && ((p.data[2] & 0xFF) == 116) && ((p.data[3] & 0xFF) == 97)
            && ((p.data[4] & 0xFF) == 121) && ((p.data[5] & 0xFF) == 103)) {
            txtVypis.setText("úspěch odpověď od cíle");
            trasa.add(p);
            lvPakety.getItems().add(p.toString());
            break;
        }
    } else {
        //cizi paket, zadna akce
    }
}
}

```

7.3.2 Načtení ze souboru

Pakety zachycené jakoukoli aplikací využívající knihovnu Jpcap a uložené pomocí Jpcap.Writer mohou být načteny do aplikace a **IP** adresy ze zachycených paketů mohou být použity pro trasování dalších uzlů v síti. Pro načtení ze souboru se využívá knihovní funkce *JpcapCaptor.openFile(String cesta)*.

```

public ArrayList<Packet> nactiZeSouboru() {
    ArrayList<Packet> packety = new ArrayList<>();
    JFileChooser fc = new JFileChooser();
    fc.showOpenDialog(fc);
    String filePath = fc.getSelectedFile().getPath();
    a = true;
    try {
        captor = JpcapCaptor.openFile(filePath);
        do {
            packet = captor.getPacket();
            if ((packet.caplen == 0) && (packet.sec == 0) && (packet.usec == 0)) {
                a = false;
            } else {
                packety.add(packet);
            }
        } while (a == true);
    } catch (java.io.IOException ioe) {
    }
    return packety;
}

```

7.3.3 Ruční editace databáze

Slouží pro doplnění informací jako fyzická adresa, popis uzlu (výrobce, majitel, typ zařízení, atd.). Odebrání uzlů, které nejsou aktivní, po změnách v síti (například stěhování serverů seznam.cz – příloha D). Odebrání uzlů s privátními adresami, které se mohou měnit.

7.4 Uložení trasy do databáze

Po zachycení trasy jsou pakety typu **ICMP** time exceeded a echo response uloženy v seznamu *List<ICMPPacket> trasa*, nejprve je uložen výchozí bod a je načteno jeho *ID*.

```

if (vychozi_Bod == null) { //pokud není v databázi uložit
    String SQL = "insert into rozhrani(ipadresa) values ('" + strip + "')";
    try {
        ResultSet rs = c.createStatement().executeQuery(SQL);
        rs.close();
    } catch (SQLException ex) {
        Logger.getLogger(FXMLDocumentController.class.getName()).log(Level.SEVERE, null, ex);
    }
    nactiRozhrani();
    SQL = "select s_rozhraniid.currval from dual";
    try {
        ResultSet rs = c.createStatement().executeQuery(SQL);
        while (rs.next()) {
            vysledek = rs.getString(1);
            idVychozihobodu = Integer.valueOf(rs.getString(1));
        }
    } catch (SQLException ex) {
        Logger.getLogger(FXMLDocumentController.class.getName()).log(Level.SEVERE, null, ex);
    }
}

```


Poté jsou postupně v cyklu procházeny jednotlivé pakety, z nich je vybírána zdrojová adresa (směrovač, který zahodil paket **ICMP** echo request), tato adresa je hledána v databázi. Pokud adresa není v databázi, je uložena a následně je uloženo spojení mezi touto adresou a adresou z předešlého paketu. Pokud již adresa v databázi je, prochází se tabulka *predchudci* pro zjištění spojení, pokud spojení neexistuje, je uloženo do databáze.

```
boolean nalezeno = false;
for (int i = 0; i < tv.getItems().size(); i++) {
    if (((String) tc.getCellData(i)).equals(strip)) {
        id = Integer.valueOf((String) ((TableColumn) tv.getColumns().get(0)).getCellData(i));
        nalezeno = true;
    }
}
int IDaktualni = -1;
if (!nalezeno) { //pokud nenalezeno v databázi uložit
    String SQL = "insert into rozhrani(ipadresa) values ('" + strip + "')";
    provedSQL(SQL);
    nactiRozhrani();
    SQL = "select s_rozhraniid.currval from dual"; //načtení posledního uloženého id
    try {
        try (ResultSet rs = c.createStatement().executeQuery(SQL)) {
            while (rs.next()) {
                IDaktualni = Integer.valueOf(rs.getString(1));
            }
        }
    } catch (SQLException ex) {
        Logger.getLogger(FXMLDocumentController.class.getName()).log(Level.SEVERE, null, ex);
    }
    //uložení spojení na předchůdce
    SQL = "insert into predchudci(id, idpredchudce, vychozibod) values ('"
        + IDaktualni + "', '" + IDPredchudce + "', '" + idVychozihoBodu + "')";
    provedSQL(SQL);
    IDPredchudce = IDaktualni;
    nactiRozhrani();
}
```

```
TableColumn tcid = (TableColumn) tv.getColumns().get(0);
TableColumn tcidpredchudce = (TableColumn) tv.getColumns().get(4);
nalezeno = false;
for (int i = 0; i < tv.getItems().size(); i++) {
    if (!"".equals((String) tcidpredchudce.getCellData(i))) {
        if ((Integer.valueOf((String) tcid.getCellData(i)) == id)
            && (Integer.valueOf((String) tcidpredchudce.getCellData(i)) == IDPredchudce)){
            nalezeno = true;
        }
    }
}
if (!nalezeno) { //pokud není spojení nalezeno uložit spojení
    String SQL = "insert into predchudci(id, idpredchudce, vychozibod) values ('"
        + id + "', '" + IDPredchudce + "', '" + idVychozihoBodu + "')";
    provedSQL(SQL);
    nactiRozhrani();
}
```

7.5 Uložení paketů do souboru

Knihovna Jpcap poskytuje funkce pro ukládání zachycených paketů do souboru. Konkrétně se jedná o třídu Jpcap.Writer. Ta do vytvořeného souboru ukládá pakety tak, aby je bylo možné znovu načíst a provádět analýzu kdykoliv, nejen ihned po zachycení.

Pakety v aplikaci uložené v *ArrayList<Packet>* jsou v cyklu ukládány do vybraného souboru pomocí metody *writer.writePacket(p)*.

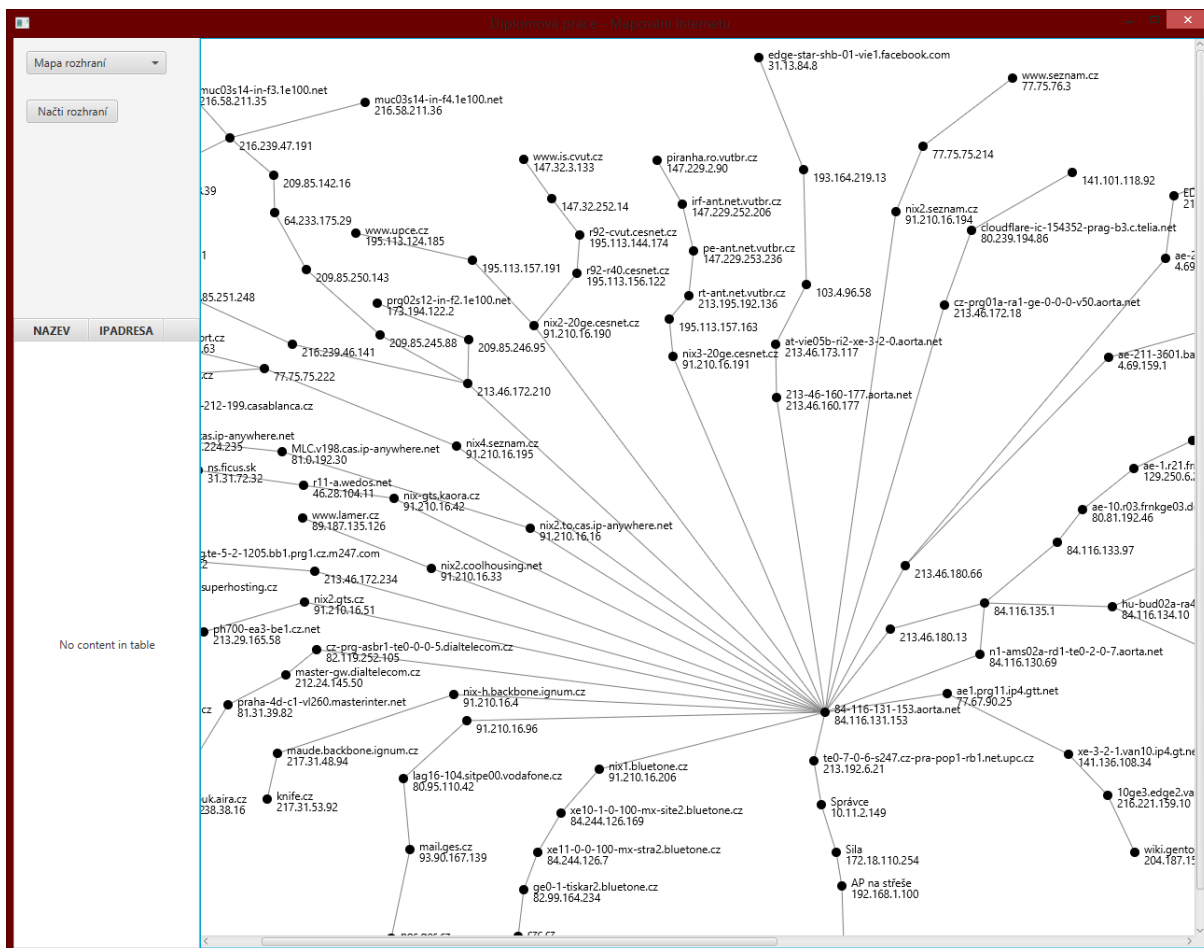
```
public void UlozDoSouboru(ArrayList<Packet> packets) {
    if (packets != null) {
        JFileChooser fc = new JFileChooser();
        fc.showSaveDialog(fc);
        File f = fc.getSelectedFile();
        try {
            JpcapWriter writer = JpcapWriter.openDumpFile(captor, f.getPath());
            for (Packet p : packets) {
                writer.writePacket(p);
            }
            writer.close();
        } catch (java.io.IOException ioe) {
        }
    }
}
```

7.6 Ovládání aplikace

V levé části okna je ovládací panel, na něm je umístěn *ChoiceBox* s výběrem ze čtyř možností: mapa rozhraní, editace databáze, zachytávač paketů, mapa s reálným podkladem. Dle zvolené možnosti se mění tlačítka a ovládací prvky v levé části, i obsah na hlavním pravém panelu, který se zvětšuje a zmenšuje dle velikosti okna.

7.6.1 Virtuální mapa rozhraní

Při volbě mapa rozhraní se zobrazuje mapa uzlů sítě na bílý podklad, s uzly je možno pohybovat tažením myši a odkládat je do tabulky v levé části obrazovky pro dosažení lepšího zobrazení sledované části sítě. Na obrázku 24 je ukázka virtuální mapy zobrazené v aplikaci.



Obrázek 24 Ukázka okna aplikace se zvolenou virtuální mapou

7.6.2 Editace databáze

Pro editaci databáze (Obrázek 25) je v levé části *ChoiceBox* na výběr tabulky z databáze, pro editaci stačí dvojklik na buňku, upravit text a potvrdit klávesou Enter. Sloupec *ID* nelze editovat neboť se jedná o primární klíč databáze a na jeho hodnotě jsou závislé prvky z dalších tabulek. Dále jsou v levé části tlačítka pro přidání a odebrání řádku z databáze. Vrchní tabulka slouží pro zobrazení vybrané tabulky, spodní pro doplňující informace (pro rozhraní jeho předchůdce, pro adresy se zobrazí všechna rozhraní na dané adrese, pro směrovače se zobrazí všechna rozhraní tohoto směrovače).

Protože většina dnešních sítí pro připojení koncových uživatelů používá **DHCP** server (automatické přidělování adres), **IP** adresa počítače se může po určitém čase změnit, z tohoto důvodu je zde přítomno nastavení výchozího bodu (**IP** adresa počítače ze kterého se provádí mapování), stačí vybrat jeden řádek z databáze a kliknout na tlačítko *Nastav výchozí bod* zrušení výběru výchozího bodu se provede kliknutím na tlačítko *Zrušit výběr výchozího bodu*.

```
@FXML
private void nactiVychozíBod() {
    if ("rozhraní".equals(nactenaTabulka)) {
        //cislo vybraného radku v tabulce
        int index = tv1.getSelectionModel().getSelectedIndex();
        //sloupec s ID
        TableColumn tc = (TableColumn) tv1.getColumns().get(0);
        //nastavení výchozího bodu
        ovladacDB.setVychozíBod(Integer.valueOf((String) tc.getCellData(index)));
    }
}
```

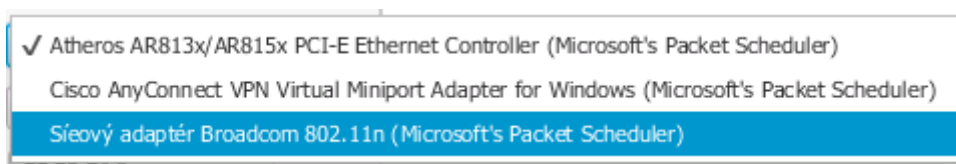
ID	NAZEV	IPADRESA	POPIS	ADRESA	SMEROVAC	X	Y	XMAPA	YMAPA	Počet předchůdce
16	PC doma	192.168.1.116				559	944	195	288	0
17	AP na střeše	192.168.1.100				500	777	298	162	1
18		172.18.110.254				433	692	481	201	1
19		10.11.2.149				439	622	501	352	1
20		213.192.6.21				517	531			1
21		84.116.131.153				656	406			1
22		91.210.16.194				539	311			1
23		77.75.75.214				438	248			1
24	www.seznam.cz	77.75.76.3				348	221			1
25		91.210.16.190				743	305			1
26		195.113.157.191				826	223			1
27	www.upce.cz	195.113.124.185				878	163			1
31		91.210.16.33				432	394			1
32	www.lamer.cz	89.187.135.126				277	348			1

ID	NAZEV	IPADRESA	POPIS	ADRESA	SMEROVAC	X	Y	XMAPA	YMAPA
18		172.18.110.254				433	692	481	201

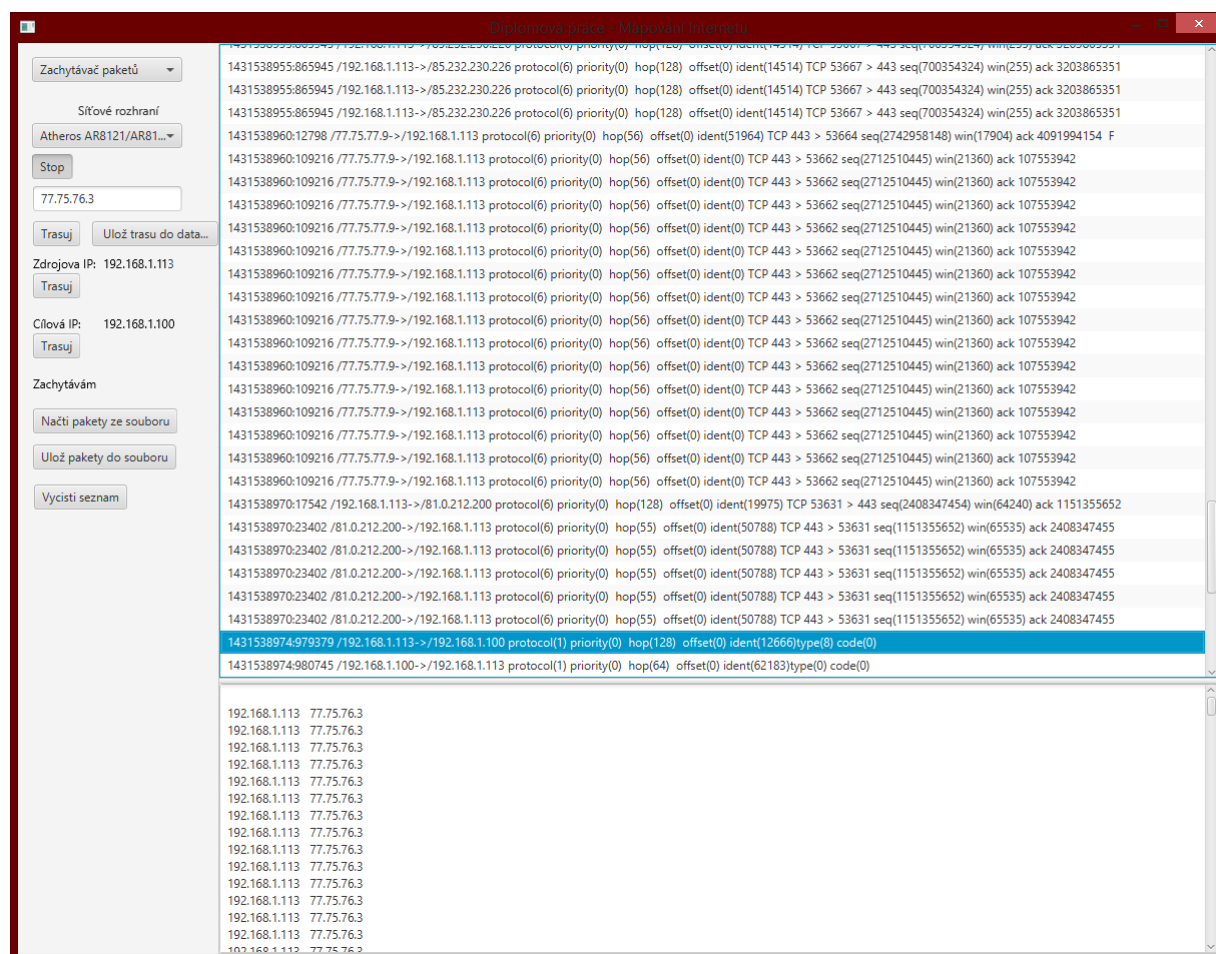
Obrázek 25 Ukázka okna aplikace se zvolenou editací databáze

7.6.3 Zachytávač paketů

Na obrázku 27 je zobrazeno zachytávání paketů v aplikaci. V levé části nahoře je výběr síťového rozhraní na kterém se bude provádět zachytávání, případně odesílání paketů pro tvorbu trasy (detail výběru na obrázku 26). Dále je zde tlačítko pro spuštění a vypnutí zachytávání. Textbox pro vložení **IP** adresy, tlačítko pro spuštění trasování a uložení trasy do databáze. Po kliknutí na jeden paket v *ListView* se v levé části zobrazí **IP** adresy zdroje (na obrázku 192.168.1.113) a cíle (na obrázku 192.168.1.100) s možností trasování. Jako poslední zde jsou tlačítka pro uložení paketů do souboru, načtení ze souboru a vymazání paketů.



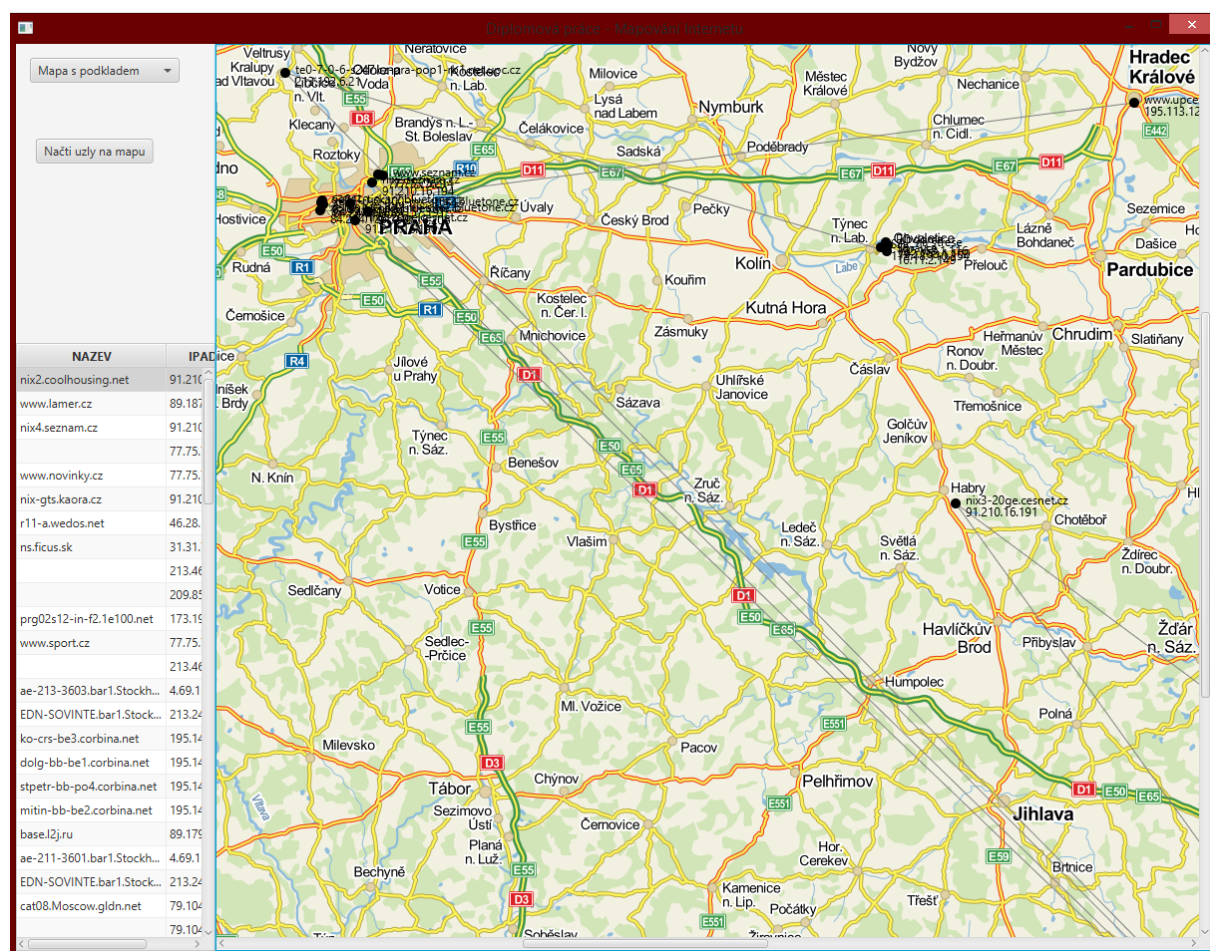
Obrázek 26 Výběr síťového rozhraní pro zachytávání



Obrázek 27 Ukázka okna aplikace se zachytáváním paketů

7.6.4 Reálná mapa rozhraní

Při volbě reálná mapa rozhraní se zobrazuje mapa uzlů sítě na mapový podklad, konkrétně mapu České republiky[15], s uzly je možno pohybovat tažením myši, tak aby poloha na mapě odpovídala skutečné poloze, také je možno uzly odkládat do tabulky v levé části obrazovky. Ukázka je na obrázku 28.



Obrázek 28 Ukázka okna aplikace s reálným mapovým podkladem

8. Vytvoření mapy

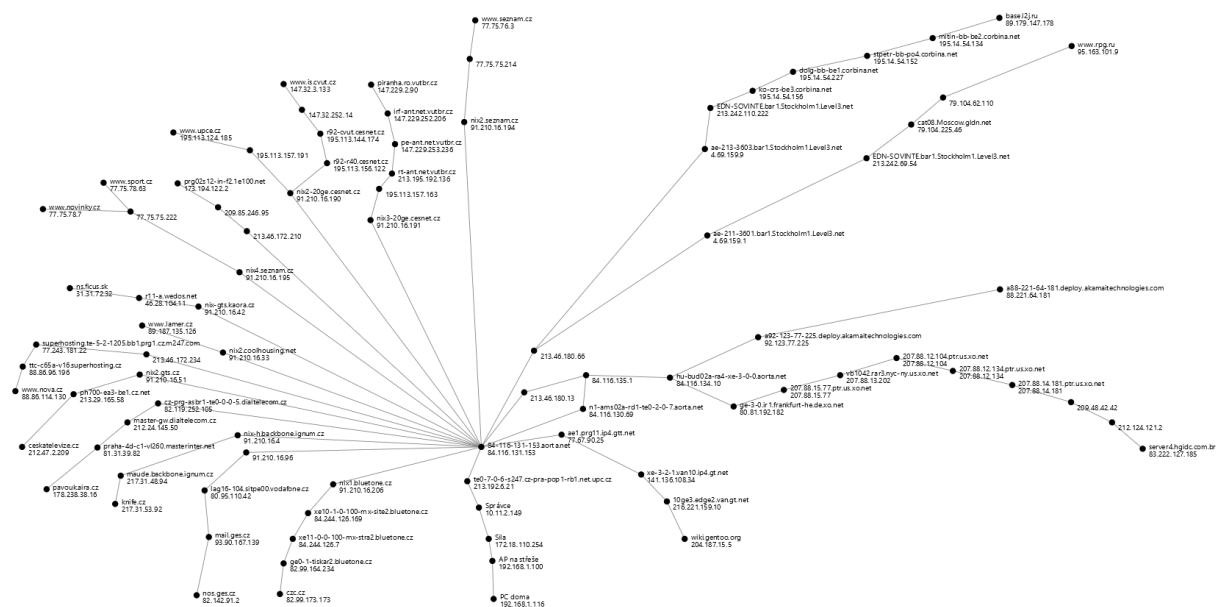
Každé rozhraní je uloženo v databázi pouze jednou, informace o něm jsou uloženy v tabulce *rozhrani*, sousední rozhraní jsou uloženy pomocí M:N spojení v tabulce *predchudci*. Do mapy se ukládá pouze rozhraní (jedno rozhraní síťového uzlu: směrovače, serveru, počítače atd.), protože pomocí analýzy paketů **ICMP** není možné určit směrovač a jeho rozhraní, ani k jednomu rozhraní zjistit další rozhraní ze stejného směrovače.

Mapu tvoří pouze **IPv4** adresy, protože bez přímého přístupu k **IPv6** síti by bylo mapování **IPv6** uzlů problematické kvůli překladu adres z **IPv6** na **IPv4**. Nicméně sloupec *IPadresa* v databázi je dostatečně velký i pro uložení **IPv6** adresy

8.1 Virtuální mapa rozhraní

Mapa rozhraní, která se vykresluje na bílou plochu, uzly lze posunovat a odebírat z plochy pro dosažení lepší vizualizace zkoumané části sítě. Pro vykreslení slouží sloupce *X* a *Y* z tabulky *rozhrani* (poloha *x* a *y* na zobrazovacím panelu).

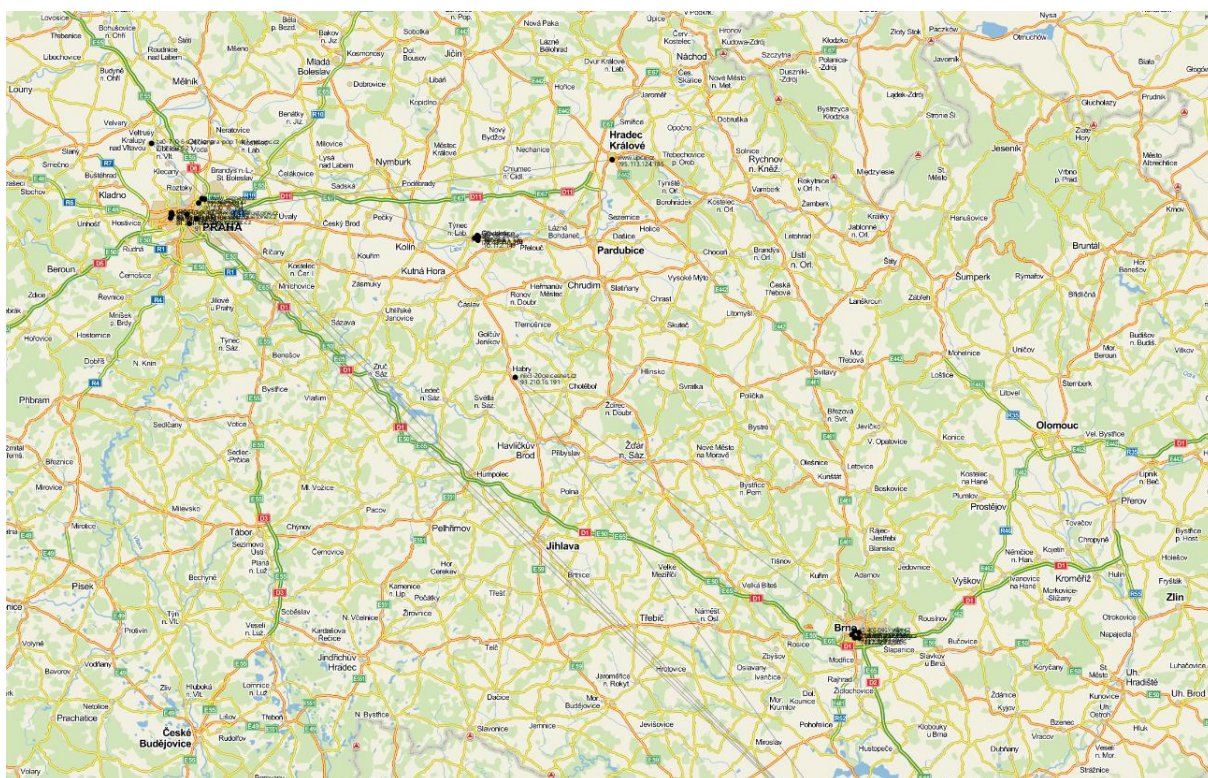
Ukázka na obrázku 29 a v příloze B zvětšená. Většina uzlů pochází z trasování českých webových stránek, některé ruské nebo americké.



Obrázek 29 Virtuální mapa rozhraní

8.2 Mapa rozhraní s reálným mapovým podkladem

Mapa se vykresluje na reálnou mapu (mapa České republiky). Zobrazuje reálnou pozici rozhraní a rozmístění prvků v síti (správnost pozice závisí na údajích zjištěných z **IP** adresy vyhledáváním v databázích vlastníků **IP** adres a na Internetu). Ukázka je na obrázku 30 a v příloze C.



Obrázek 30 Reálná mapa rozhraní

Závěr

Tato diplomová práce měla za cíl zjistit možnosti mapování Internetu a vytvořit vlastní program pro zachytávání a odesílání paketů, dále pro tvorbu mapy sítě, její uložení do databáze a zobrazení. V teoretické části byly popsány využití síťové protokoly, síťová zařízení, vybrané mapy Internetu, a technologie Jpcap pro práci se sítí v jazyce Java. Tyto teoretické poznatky byly následně využity v praktické části při tvorbě aplikace pro trasování, ukládání do databáze a vykreslování mapy na obrazovku.

Již během programování bakalářské práce jsem narazil na problém, že přestaly fungovat webové stránky knihovny Jpcap, kterou jsem využíval. Naštěstí jsem měl knihovnu i některé vzory stažené a dalo se bez dokumentace obejít.

Jako další rozšíření aplikace by bylo vhodné doplnit DNS překlad adres na jména (vytvoření, odeslání DNS paketu a čekání na odpověď), dále měření odezvy. Případně využití aplikace pro tvorbu větší mapy s více uzly. A také vylepšení zobrazení, například zvýraznění cesty v síti, barevné oddělení vnitřních sítí, páteřních sítí a jiných, nebo obarvení uzlů podle země, ke které uzel patří.

Aplikace pro tvorbu mapy a následně vytvořená mapa mají nejrůznější využití. Jako první je zmapování Internetu, zjištění kudy prochází pakety například při načtení domovské stránky. Toto je možné využít v oblasti bezpečnosti, po zjištění, že nějaká část sítě je riziková, odesílat pakety jinou trasou, nebo využít šifrování. Mapu sítě lze také využít jako obranu proti DDos útokům. Pokud by útoky přicházely z jedné části sítě (například jedna země) tak by bylo možné spojení z této země zablokovat, nebo alespoň snížit přenosovou rychlost útočníka pod přenosovou rychlost serveru aby mohl odpovídat na všechny dotazy. Dále lze využít teorii grafů (protože síť je v podstatě graf) pro vylepšení směrování, aby pakety neputovaly přes síť „zkratkou“ (například spojení z Prahy do Pardubic přes Francii), nebo také pro doplnění nových připojení v místech kde je síť příliš zatížená. Také lze simulovat, jak ovlivní funkčnost sítě například výpadek některého páteřního směrovače a připojit tak sledovaný server například na 3 různá místa v síti, aby nedošlo k výpadku služby ani při výpadku jednoho připojení.

Na druhou stranu je mapa sítě může být mapa sítě zneužitelná jako průzkum před nějakým útokem, například DDos útokem, tvorba botnetu (síť počítačů, která posílá velké množství požadavků na server, tak aby tyto požadavky server přetížily a ten přestal odpovídat) tak aby mohl zasáhnout kterékoliv místo v síti (je potřeba velká přenosová

rychlost), nebo útoku v síti vedeném tak aby trasa útoku vedla místy kde je minimální obrana (firewally, detekční a prevenční systémy atd.).

Tvorba diplomové práce pro mě byla přínosná, při tvorbě aplikace jsem si vyzkoušel připojení k databázi a novou technologii JavaFX pro tvorbu grafického rozhraní, která je mnohem lepší a modernější než původní Swing.

Od počátku tvorby jsem měl na papíře nakreslenou jednoduchou mapu s 21 uzly jako ukázkou jak by měla vypadat hotová mapa. Díky této mapě jsem mohl pozorovat změny v síti (stěhování serverů firmy seznam.cz), měl jsem zaznamenány trasy ke dvěma serverům a postupně se měnily IP adresy před serverem a následně jeden ze serverů přestal odpovídat úplně. Stav sítě i změny jsou v příloze D.

Literatura

Knižní zdroje

- [1] PECINOVSKÝ, Rudolf. Návrhové vzory: návrh a tvorba aplikací. Vyd. 1. Brno: Computer Press, 2008, 527 s. Programování. ISBN 978-80-251-1582-4.
- [2] JONES, Meilir, Základy objektově orientovaného návrhu v UML: návrh a tvorba aplikací. Vyd. 1. Praha: Grada, 2001, 367 s. Programování. ISBN 80-247-0210-X.
- [3] PUŽMANOVÁ, Rita. TCP/IP v kostce: návrh a tvorba aplikací. 1. vyd. České Budějovice: Kopp, 2004, 607 s. Programování. ISBN 80-7232-236-2 .

Internetové zdroje

- [4] POSTEL, J. Request for Comments: 792 INTERNET CONTROL MESSAGE PROTOCOL [online] [cit. 2013-06-15]. Dostupné z <http://www.ietf.org/rfc/rfc792.txt>
- [5] FUJII, Keita. JPCAP library [online] [cit. 2012]. Od roku 2013 nedostupné. Původní adresa: <http://www.netresearch.ics.uci.edu/kfujii/jpcap/doc/index.html>
- [6] Abhaya S Induruwa IPv6 Header Format [online] 2001-12-16 [cit. 2013-07-15] Dostupné z: http://agenda.ictp.trieste.it/agenda_links/smr1335/networking/node35.html
- [7] Programming tutorials and source code examples [online] 2009-2012 [cit. 2013] Dostupné z: <http://java2s.com/>
- [8] TILLMAN, Karen. How Many Internet Connections are in the World? Right. Now. [online] 2013-06-29 [cit. 2015-05-08] Dostupné z: <http://blogs.cisco.com/news/cisco-connections-counter>
- [9] MAHLKNECHT. Greg, Greg's Cable Map [online] 2014-10-26 [cit. 2015-04-21] Dostupné z: <http://www.cablemap.info/>
- [10] CHESWICK, William. BURCH, Hal. The Internet Mapping Project [online] [cit. 2015-05-05] Dostupné z: <http://www.cheswick.com/ches/map/>
- [11] LUMETA. Lumeta Internet Mapping Project [online] [cit. 2015-05-05] Dostupné z: <http://internet-map.net/>
- [12] Vyhledávání v .cz doménách. [online] 2007-2015 [cit. 2015-05-10] Dostupné z: www.czdomeny.cz

- [13] The AWT Focus Subsystem [online] [cit. 2015-04-20] Dostupné z <http://docs.oracle.com/javase/8/docs/api/java/awt/doc-files/FocusSpec.html>
- [14] A Brief History of JavaFX [online] 2007-2015 [cit. 2015-04-20] Dostupné z <http://what-when-how.com/javafx-2/a-brief-history-of-javafx-getting-a-jump-start-in-javafx/>
- [15] Mapa České republiky [online] [cit. 2014-10-15] Dostupné z www.mapy.cz
- [16] SMITH, Jeff. Social Networking 9.0 – Reboot (History of Social Networking). [online] 2011-06-29 [cit. 2015-03-18] Dostupné z:
- [17] Arpanet [online] [cit. 2015-03-18] Dostupné z: <http://malarky.udel.edu/~dmills/pic/maps/arpanet79a.jpg>
- [18] The Internet Map [online] [cit. 2015-02-23] Dostupné z: <http://internet-map.net/>
- [19] DB-IP [online] [cit. 2015-05-09] Dostupné z: <https://db-ip.com/>
- [20] LYON, Barrett. The OPTE Project. [online] 2014 [cit. 2015-05-14] Dostupné z: <http://www.opte.org>
- [21] IANA/ICANN [online] [cit. 2015-05-14] Dostupné z: <http://www.iana.org>

Příloha A - Obsah CD

Zdrojový kód aplikace (NetBeans JavaFXML projekt)

Instalátor knihovny Jpcap

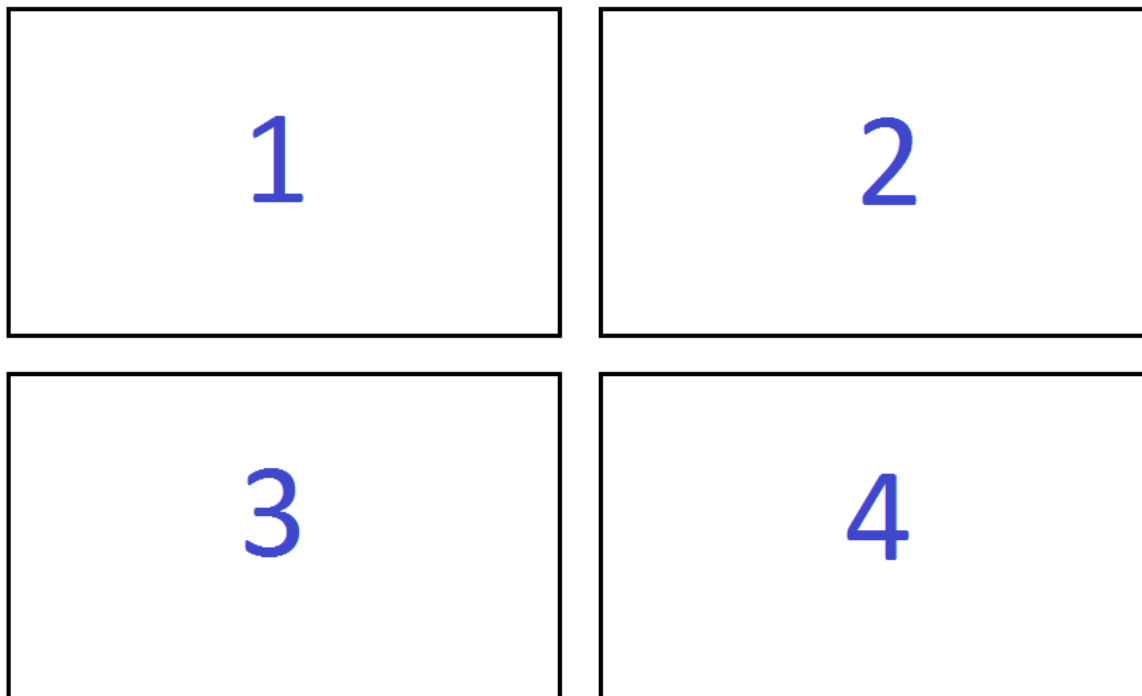
Instalátor knihovny Winpcap

SQL skripty pro tvorbu tabulek

SQL skripty pro naplnění tabulek mapou

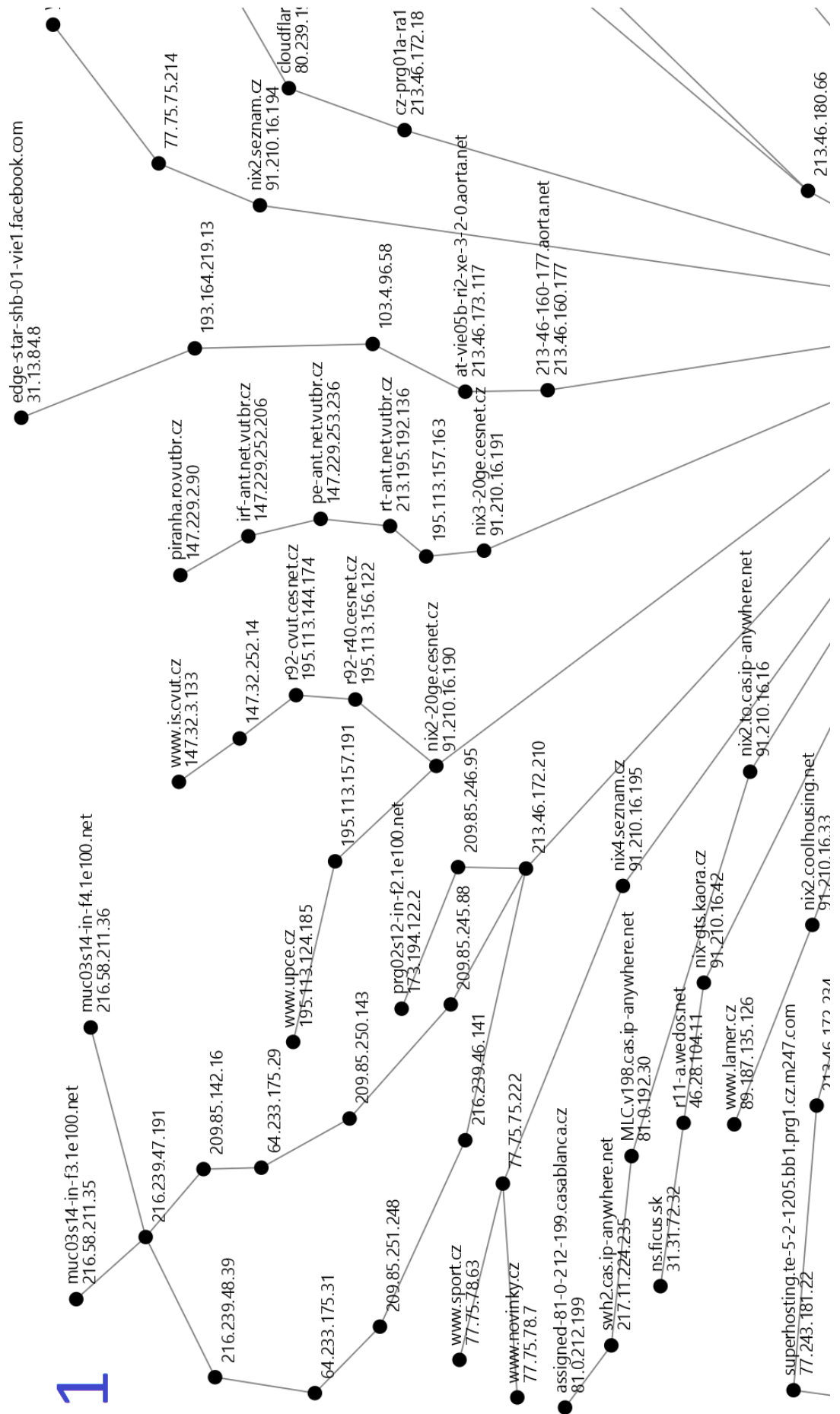
Příloha B – Virtuální mapa rozhraní

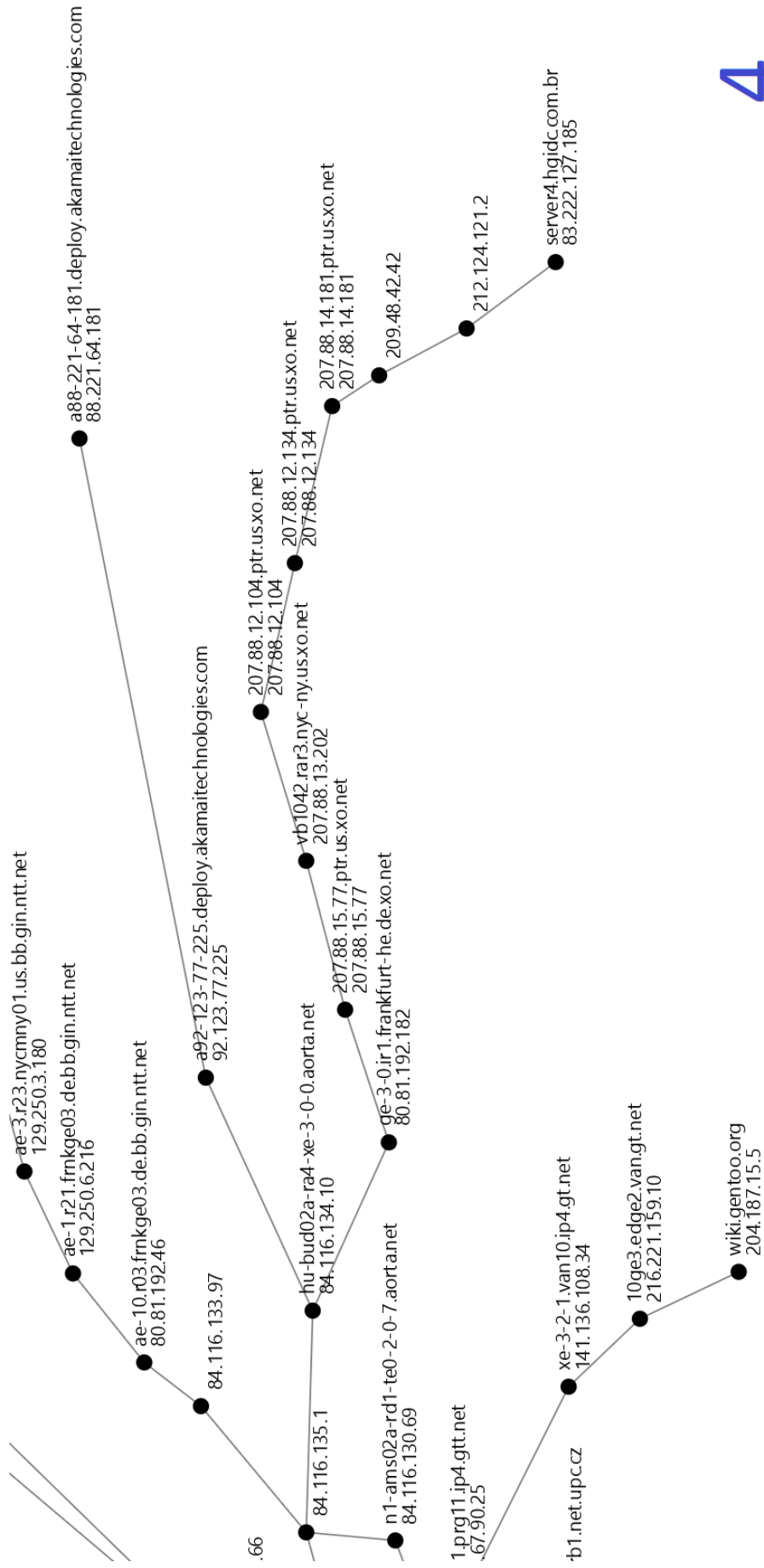
Mapa se skládá ze 4 volných listů vložených na šířku, na obrázku 31 je znázorněno rozložení listů



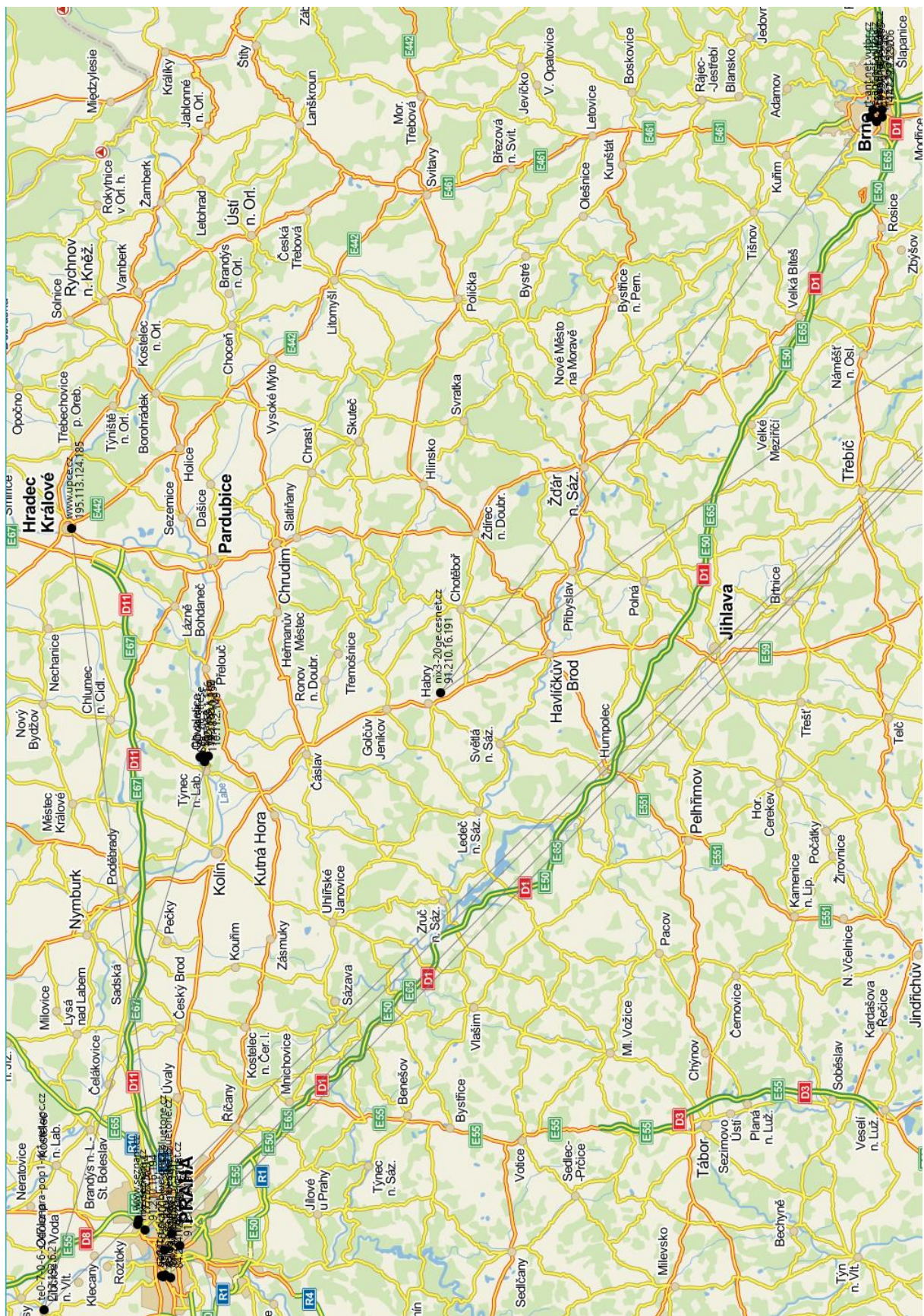
Obrázek 31 Rozložení listů virtuální mapy

1





Příloha C – Reálná mapa rozhraní

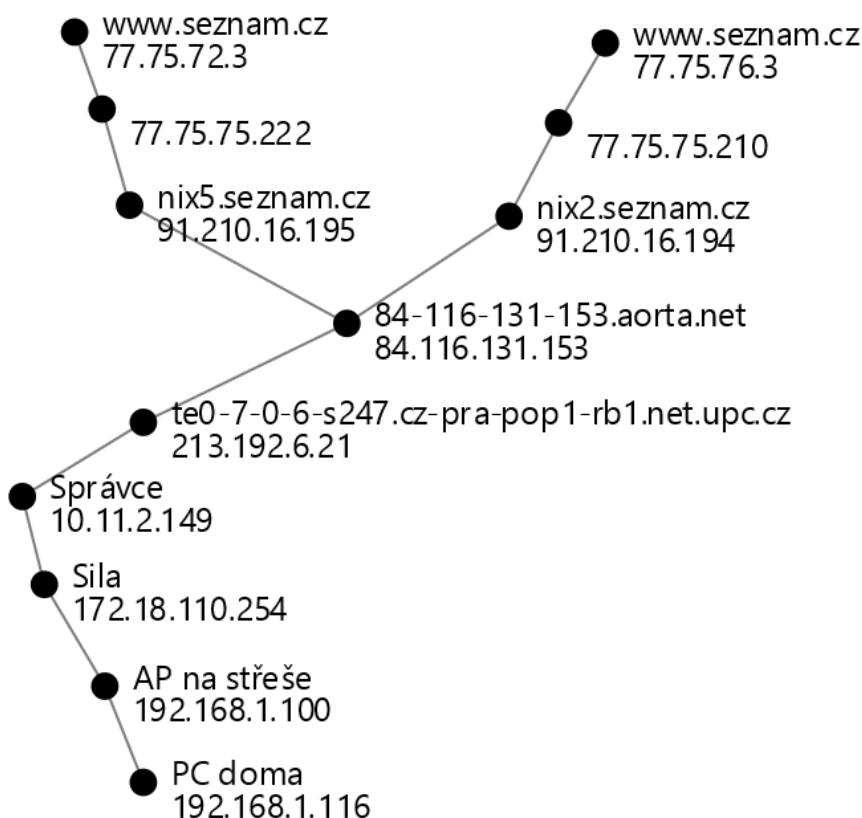


Příloha D – Stěhování serverů firmy seznam.cz

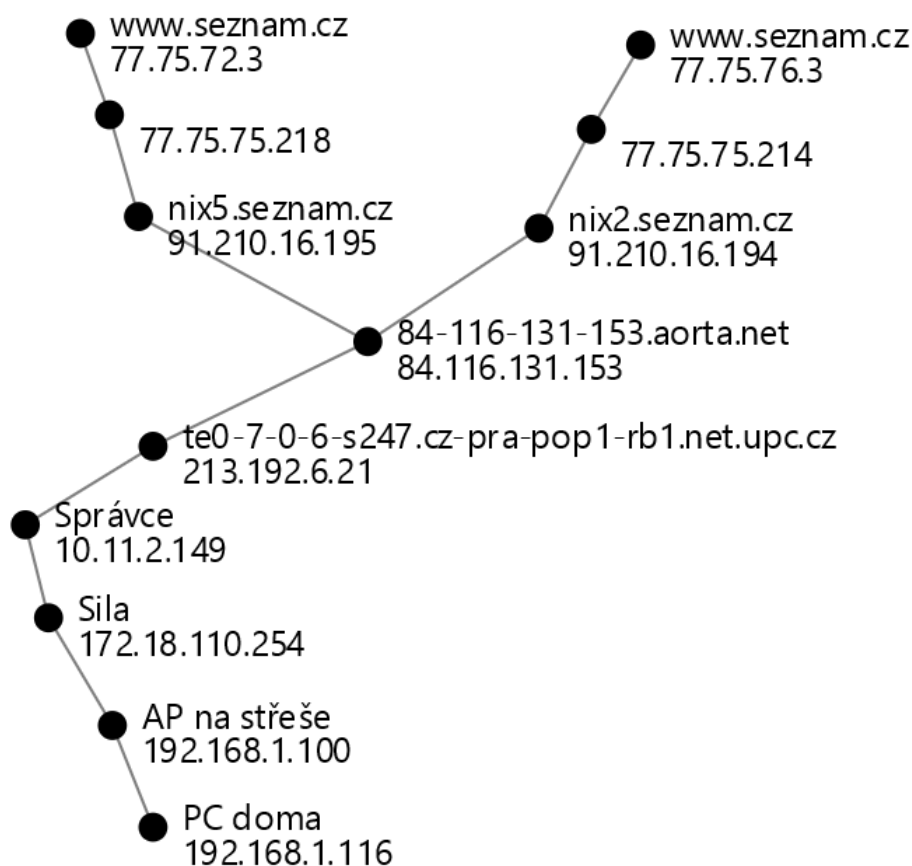
V roce 2015 seznam uvádí do provozu vlastní nové datové centrum Kokura v Praze Horních Počernicích s náklady okolo 200 milionů korun, od dubna bude přesouvat 1300 ze tří tisíc serveru. Stěhování potrvá dva měsíce a provoz služeb Seznamu nesmí být nijak omezen.

V mapě uzlů se přesun projevil změnou IP adres směrovačů, a (nejspíš) fyzickým přemístěním koncových serverů, které při trasování není vidět. Následující mapy ukazují dva servery s IP adresami 77.75.76.3 a 77.75.72.3, které byly zaznamenány již v září 2014.

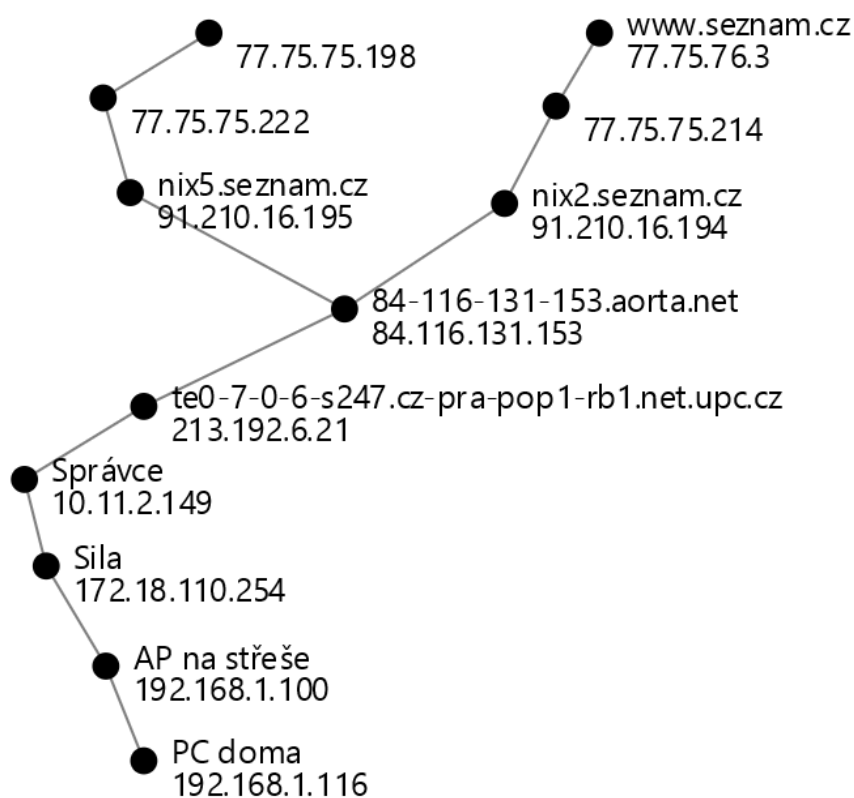
Na obrázku 22 je původní stav trasy k serverům www.seznam.cz ze září 2014, na dalším obrázku (23) je stav po první změně (duben 2015) kdy došlo ke změně IP adres směrovačů před servery, na obrázku 24 je stav z května 2015 kdy server na IP adrese 77.75.76.3 stále funguje, ale server s IP adresou 77.75.72.3 neodpovídá.



Obrázek 32 Mapa trasy k serverům seznam.cz ze září 2014



Obrázek 33 Mapa trasy k serverům seznam.cz z dubna 2015



Obrázek 34 Mapa trasy k serverům seznam.cz z 12. května 2015