



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**

FACULTY OF INFORMATION TECHNOLOGY

**ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ**

DEPARTMENT OF COMPUTER SYSTEMS

**GENERÁTOR IPV6 PREFIXŮ**

IPV6 PREFIX GENERATOR

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**VEDOUCÍ PRÁCE**

SUPERVISOR

**VOJTĚCH OBRUSNÍK**

**JIŘÍ MATOUŠEK, Ing.**

BRNO 2018

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

Ústav počítačových systémů

Akademický rok 2017/2018

## Zadání bakalářské práce

Řešitel: **Obrusník Vojtěch**  
Obor: Informační technologie  
Téma: **Generátor IPv6 prefixů**  
**IPv6 Prefix Generator**  
Kategorie: Počítačové sítě

**Pokyny:**

1. Nastudujte princip generování IPv6 prefixů popsany v článku: K. Zheng, B. Liu. *V6Gene: A Scalable IPv6 Prefix Generator for Route Lookup Algorithm Benchmark*. In Proceedings of the 20th International Conference on Advanced Information Networking and Applications (AINA'06), 2006.
2. Proveďte implementaci nastudovaného přístupu v jazyce Python.
3. Seznamte se s pravidly pro přidělování IPv6 adres.
4. Navrhněte vlastní způsob generování IPv6 prefixů, který bude využívat znalost pravidel pro přidělování IPv6 adres.
5. Navržený způsob generování implementujte v jazyce Python.
6. Ověřte vlastnosti implementovaného generátoru a srovnajte je s vlastnostmi přístupu prezentovaného v článku z bodu 1 zadání.
7. V závěru diskutujte možnosti dalších vylepšení vámi navrženého a implementovaného generátoru.

**Literatura:**

- K. Zheng, B. Liu. *V6Gene: A Scalable IPv6 Prefix Generator for Route Lookup Algorithm Benchmark*. In Proceedings of the 20th International Conference on Advanced Information Networking and Applications (AINA'06), 2006.
- RIPE Network Coordination Centre. *IPv6 Address Allocation and Assignment Policy* [online]. Rev. 21 May 2012 [cit. 2012-10-31]. Available at URL: <<http://www.ripe.net/ripe/docs/ipv6policy.html>>.
- American Registry for Internet Numbers. *ARIN Number Resource Policy Manual* [online]. Rev. 31 July 2012 [cit. 2012-10-31]. Available at URL: <<https://www.arin.net/policy/nrpm.html>>.
- APNIC. *IPv6 address allocation and assignment policy* [online]. Rev. 19 August 2012 [cit. 2012-10-31]. Available at URL: <<http://www.apnic.net/policy/ipv6-address-policy>>.

Pro udělení zápočtu za první semestr je požadováno:

- Splnění bodů 1 až 3 zadání.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Matoušek Jiří, Ing.**, UPSY FIT VUT

Datum zadání: 1. listopadu 2017

Datum odevzdání: 16. května 2018

**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

Fakulta informačních technologií

Ústav počítačových systémů a sítí

602 00 Brno, Božetěchova 2



prof. Ing. Lukáš Sekanina, Ph.D.  
vedoucí ústavu

## Abstrakt

Nastávající přechod na IPv6 technologii s sebou přináší požadavek na existenci efektivních vyhledávacích algoritmů směrovacích tabulek, jenž by byly přizpůsobeny práci s délkami IPv6 prefixů. Vývoj takových algoritmů však vyžaduje množství testovacích dat, která v současné době neexistují. Tato práce se zaměřuje na návrh a implementaci nástroje, který by byl schopen testovací data generovat. Vytvoření generátoru předcházelo nastudování pravidel přidělování IPv6 prefixů v reálném prostředí a také analýza stávajících prefixových sad. K ověření funkčnosti generátoru byly použity metody srovnání rozložení délek a hodnot bitů prefixu mezi reálnou a uměle získanou sadou dat.

## Abstract

Incoming deployment of IPv6 technology in larger scale requires existence of effective route lookup algorithms suitable of manipulation with long IPv6 prefixes. Development of such algorithms is dependent on an huge amount of test data sets, which do not exist at the moment. This paper is focused on design and implementation of a tool, that would be able to generate such test data sets. Study phase of IPv6 address allocation polices and analysis of currently available pieces of prefix sets took place before the implementation. Validation of results was performed by bit value and prefix length scatter measurement.

## Klíčová slova

generator, IPv6 adresa, IPv6 prefix, přidělování IPv6 adres, V6Gene

## Keywords

generator, IPv6 address, IPv6 prefix, IPv6 address assignment, V6Gene

## Citace

OBRUSNÍK, Vojtěch. *Generátor IPv6 prefixů*. Brno, 2018. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Jiří Matoušek, Ing.

# Generátor IPv6 prefixů

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Jiřího Matouška. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....  
Vojtěch Obrusník  
16. května 2018

## Poděkování

Zde bych rád poděkoval Ing. Jiřímu Matouškovi především za jeho odborné rady, obětavý přístup a veškerý čas a úsilí, které vložil do konzultací při vedení mé práce.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
<b>2</b>	<b>IPv6 adresy a směrování v IPv6 sítích</b>	<b>5</b>
2.1	IPv6 adresa . . . . .	5
2.1.1	Typy . . . . .	5
2.1.2	Model adresování . . . . .	5
2.1.3	Textová reprezentace . . . . .	6
2.1.4	Adresový prefix . . . . .	6
2.2	Směrování v IPv6 sítích . . . . .	7
2.2.1	Průběh směrování na směrovači . . . . .	7
2.2.2	Směrovací protokoly . . . . .	7
2.2.3	Směrovací tabulka . . . . .	8
<b>3</b>	<b>Přidělování IPv6 adres</b>	<b>9</b>
3.1	Základní pojmy . . . . .	9
3.2	Cíle řízení přidělování IPv6 adres . . . . .	10
3.3	Hierarchie internetových registrátorů . . . . .	11
3.4	Pravidla přidělování IPv6 adres . . . . .	11
3.4.1	IANA . . . . .	12
3.4.2	RIR . . . . .	12
3.4.3	LIR/NIR . . . . .	14
<b>4</b>	<b>Současný stav</b>	<b>15</b>
4.1	Přehled přístupů . . . . .	15
4.1.1	Non-random generator for IPv6 tables . . . . .	15
4.1.2	V6Gene . . . . .	16
<b>5</b>	<b>Generátor IPv6 prefixů dle V6gene</b>	<b>18</b>
5.1	Návrh . . . . .	18
5.2	Implementace . . . . .	19
5.2.1	Parametry . . . . .	19
5.2.2	Inicializace . . . . .	20
5.2.3	Generování . . . . .	20
5.2.4	Výstup . . . . .	21
5.3	Spuštění . . . . .	21
5.3.1	Vstupní data a parametrizace . . . . .	21
<b>6</b>	<b>Analýza reálných IPv6 prefixových sad</b>	<b>24</b>

6.1	Zdroje reálných dat . . . . .	24
6.2	Metodiky analýzy . . . . .	24
6.3	Provedení analýzy . . . . .	25
6.4	Časový vývoj prefixové sady . . . . .	25
6.5	Vyhodnocení . . . . .	26
<b>7</b>	<b>Návrh a implementace vlastního generátoru</b>	<b>28</b>
7.1	Požadavky . . . . .	28
7.2	Návrh . . . . .	28
7.2.1	Parametry . . . . .	28
7.2.2	Generování . . . . .	29
7.3	Implementace . . . . .	29
7.3.1	Parametry . . . . .	29
7.3.2	Inicializace . . . . .	30
7.3.3	Generování . . . . .	30
7.3.4	Výstup . . . . .	31
7.4	Spuštění . . . . .	31
<b>8</b>	<b>Srovnání výsledků</b>	<b>34</b>
8.1	Výstupy generování . . . . .	34
8.1.1	Rozložení bitů . . . . .	34
8.1.2	Rozložení délek . . . . .	34
8.1.3	Efektivita generátorů . . . . .	35
<b>9</b>	<b>Závěr</b>	<b>37</b>
	<b>Literatura</b>	<b>38</b>
<b>A</b>	<b>Obsah CD</b>	<b>40</b>

# Kapitola 1

## Úvod

Široké možnosti využití Internetu a tím podmíněná rostoucí obliba ze strany uživatelů zapříčinila rychlý a rozsáhlý nárůst počtu aplikací. Přímo souvisejícím jevem je i zvětšující se množství připojených zařízení. To má za následek postupné vyčerpávání adresového prostoru IP (Internet Protocol) verze 4 (IPv4), který je aktuálně páteřním protokolem pro komunikaci v celém Internetu. I přes značnou velikost IPv4 adresového prostoru, kde lze vytvořit až  $2^{32}$  jedinečných adres, a různé techniky aplikované za účelem dočasně problém s ubývajícím prostorem překlenou (např. NAT – Network Address Translation) došlo v únoru roku 2011 k rozdělení posledních volných adresových bloků organizací IANA (Internet Assigned Numbers Authority) jednotlivým globálním registrátorům (RIR – Regional Internet registry) [12]. Nadcházející úplné vyčerpání IPv4 adresového prostoru jednotlivých RIR je odhadováno na období mezi roky 2021 a 2022 [11].

Tento předpokládaný zlomový bod historie Internetu spustil výzkum a vývoj nového způsobu adresování síťových zařízení. Vývoj přinesl v roce 1995 první návrh nové verze adresování IP – IPv6. Hlavní předností IPv6 adresového prostoru je především jeho velikost. S délkou adresy 128 bitů je možné vytvořit o  $2^{96}$  více adres oproti IPv4. Také umožňuje např. přímé spojení dvou koncových bodů v síti bez použití NAT, nebo lepší podporu pro zajištění QoS (Quality of service). Markantní rozdíl mezi IPv4 a IPv6 s sebou však přináší i jistá úskalí.

Zásadní pro přenos informace – paketu – mezi jednotlivými uzly v Internetu je směrování. To probíhá ve směrovačích, kde je směrovacím algoritmem právě na základě IP adresy určeno výstupní rozhraní, na které bude paket odeslán. Efektivita výběru výstupního rozhraní a tudíž i rychlost přenosu dat v síti závisí na schopnosti směrovacích algoritmů vyhledat ve struktuře směrovací tabulky požadovanou informaci. Princip vyhledávání spočívá v nalezení shody nejdelší možné části začátku IP adresy s některým ze záznamů uložených ve směrovací tabulce, tzv. síťovým prefixem. Zásadním způsobem tedy ovlivňuje efektivitu vyhledávání délka, resp. velikost informace. Právě ta se v IPv6 značně liší od IPv4. Existující vyhledávací algoritmy tak nelze použít v aktuální podobě. Vytvoření nových či úprava existujících s sebou ovšem přináší nutnost testování, ke kterému jsou nezbytná testovací data. Vzhledem k rannému stádiu nasazení IPv6 technologie však neexistuje dostatečné množství reálných dat – prefixových sad – vhodných k testování.

Tato práce si klade za cíl popsat nastudování, implementaci a pokus o vylepšení jednoho z možných přístupů k získání těchto prefixových sad.

Kapitola 2 popisuje formát IPv6 adresy a proces směrování v IPv6 sítích. Kapitola 3 se věnuje problematice přidělování adresového prostoru. Se všemi předcházejícími prerekvizitami představuje kapitola 4 stávající přístupy ke generování prefixových sad. Na základě

představení poté kapitola 5 uvádí postup návrhu a implementace generátoru, který staví na principu popsaném v článku [17]. Implementaci vlastního přístupu 7 předchází kapitola 6 s možnostmi jak napříč daty sledovat závislosti a vyvozovat z nich důsledky. Kapitola 8 uvádí srovnání přístupů a diskutuje efektivitu vytvořených nástrojů. V závěru 9 je možné najít zhodnocení celého procesu a nastínění dalšího vývoje.

## Kapitola 2

# IPv6 adresy a směrování v IPv6 sítích

V této kapitole je popsána struktura IPv6 adresy, její kategorizace na základě typu a možné formáty textové reprezentace s uvedenými příklady. Dále je uveden popis podoby IPv6 prefixu a osvětlen způsob směrování datagramu v IPv6 síti.

### 2.1 IPv6 adresa

IPv6 adresa je číselný identifikátor sloužící k identifikaci jednoho či více (v závislosti na typu) síťových rozhraní. Délka identifikátoru je 128 bitů a může být následujících typů: (V této sekci je popsána adresa IP verze 6 dle [10])

#### 2.1.1 Typy

- **Unicast** — Identifikátor jednoho rozhraní. Paket odeslaný na adresu typu unicast je doručen na rozhraní identifikované touto adresou.
- **Anycast** — Identifikátor množiny rozhraní, která náleží různým uzlům. Paket odeslaný na adresu typu anycast je doručen na jedno z množiny rozhraní, které je nejbližší odesílateli dle metrik směrovacích algoritmů.
- **Multicast** — Identifikátor množiny rozhraní. Paket odeslaný na adresu typu multicast je doručen na všechna rozhraní množiny.

Na rozdíl od IPv4 adres není v IPv6 definována adresa typu broadcast, jejichž funkci přebírají adresy typu multicast.

Důležitým poznatkem, který se týká typů IPv6 adresy, je způsob rozpoznání typu adresy podle pořadí nejlevějších bitů identifikátoru takto, viz tabulka 2.1.

#### 2.1.2 Model adresování

Pro všechny typy IPv6 adres platí, že jsou přiřazeny rozhraním, ne síťovým uzlům. Adresa typu unicast odkazuje na jediné rozhraní. Jelikož každé rozhraní náleží některému síťovému uzlu, lze kteroukoliv adresu typu unicast některého rozhraní uzlu využít k identifikaci tohoto uzlu. Je vyžadováno, aby každé aktivní rozhraní mělo alespoň lokální adresu typu unicast, přičemž rozhraní může mít přiřazených více adres různého typu zároveň.

Typ adresy	Binární prefix	IPv6 reprezentace
Nedefinováno	00...0 (128 bitů)	::/128
Loopback	00...1 (128 bitů)	::1/128
Multicast	11111111	FF00::/8
Link-Local unicast	1111111010	FE80::/10
Globální unicast	(vše ostatní)	

Tabulka 2.1: Pořadí nejlevějších bitů IPv6 adresy dle typu

### 2.1.3 Textová reprezentace

Existují 3 běžné způsoby reprezentace IPv6 adresy v podobě textového řetězce, jsou to tyto:

- Preferovaná varianta zápisu je `x:x:x:x:x:x:x:x`, kde `x` reprezentuje jednu až čtyři hexadecimální číslice. Celkově tedy osm 16 bitových úseků adresy. V zápisu není třeba uvádět nuly na předcházejících pozicích v rámci skupiny `x`. V rámci skupiny je však vyžadováno uvést minimálně jednu číslici. Příklady:

```
2001:0DB8:0000:0000:0008:0800:200C:417A
2001:DB8:0:0:8:800:200C:417A
```

```
0000:0000:0000:0000:0000:0000:0000:0001
0:0:0:0:0:0:0:1
```

- Zjednodušená varianta zápisu umožňuje u adres s výskytem dlouhých skupin nulových bitů tyto zapsat zkráceně uvedením dvojice znaků dvojtečka (`::`). Zkratka `::` označuje jednu či více 16 bitových skupin nulových bitů a v rámci zápisu adresy může být použita pouze jedenkrát. Příklady:

```
2001:DB8::8:800:200C:417A
::1
```

- Alternativní varianta zápisu je vhodná zejména pro využití v kombinovaném prostředí IPv4 a IPv6. Její tvar je `x:x:x:x:x:x.d.d.d.d`, kde `x` jsou hexadecimální číslice popsané výše, které tvoří prvních šest 16 bitových skupin, `d` potom zastupuje čtveřici desítkových číslic – 8 bitových skupin – standardní reprezentace IPv4 adresy. Příklad:

```
2001:DB8::8:800:129.144.52.38
```

### 2.1.4 Adresový prefix

Textová reprezentace adresového prefixu je podobná klasickému zápisu IPv4 prefixu v CIDR (Classless Inter-Domain Routing) notaci. Zápis prefixu vypadá takto:

```
ipv6-adresa/délka-prefixu
```

příčemž

- `ipv6-adresa` — IPv6 adresa popsaná v sekci 2.1.
- `délka-prefixu` — Číslo v desítkové soustavě, které říká, jak dlouhý souvislý úsek nejlevějších bitů v adrese tvoří prefix.

Validní prefix nesmí obsahovat jedničkové bity za hranicí délky prefixu, pokud se nejedná o zkrácený sloučený zápis adresy uzlu, resp. rozhraní, a prefixu.

Příklad validního zápisu prefixu:

```
2001:0DB8:0:CD30::/60
```

Příklad nevalidního zápisu prefixu:

```
2001:0DB8::CD30/60
```

## 2.2 Směrování v IPv6 sítích

Směrování je proces určování cesty, kterou bude procházet informace mezi dvěma sítěmi. V případě sítí s přepínáním paketů, pro které byl IP navržen, probíhá směrování ve směrovačích na třetí – síťové – vrstvě dle modelu ISO OSI (International Organization for Standardization — Open Systems Interconnection).

### 2.2.1 Průběh směrování na směrovači

Základní postup směrování je totožný pro IPv4 i IPv6. Směrovačem obdrženy datagram je rozbalen a z hlavičky datagramu je vyčtena cílová IP adresa. K té jsou následným prohledáváním směrovačích tabulky nalezeny odpovídající prefixy, z nichž bude použit ten nejdelší na základě shody bitů adresy a prefixu zleva. Datagram bude znovu zabalen a odeslán na příslušné rozhraní určené nalezeným prefixem.

### 2.2.2 Směrovací protokoly

Funkce směrovacích protokolů spočívá především v dynamické analýze topologie sítě a propagací zjištěných informací k okolním směrovačům. Účelem je, aby měl každý směrovač dostatek údajů k efektivnímu výběru cesty mezi libovolnými dvěma uzly v síti. Existuje řada směrovacích protokolů, které se navzájem liší způsobem komunikace, jednotlivými metrikami hodnocení optimální cesty atd. Pro IPv6 přicházejí v úvahu následující:

#### RIPng (Routing Information Protocol next generation)

Velmi starý a jednoduchý směrovací protokol vycházející z původního protokolu RIP. Základní metrikou pro určení optimální cesty je počet skoků mezi jednotlivými směrovači. Počet skoků je omezen maximální hodnotou 15. Zaslání směrovačích tabulky sousedním směrovačům probíhá každých 30 vteřin a pravidla pro jejich aktualizaci zůstala shodná jako v původní verzi protokolu. Uplatnění pro tento protokol lze nalézt v menších či středně velkých sítích. RIPng je definován v [13].

## OSPFv3 (Open Shortest Path First)

Označením OSPFv3 je myšlena varianta protokolu OSPF pro IPv6. Oproti RIPng jde o podstatně složitější, ale zároveň výkonější, protokol. Základní metrikou pro určení optimální cesty je tzv. cena spočítaná na základě poměru šířky referenčního pásma a šířky pásma konkrétního rozhraní. Propagace směrovacích informací probíhá na základě zjištění změny v topologii. Protokol byl navržen pro práci v rozsáhlých sítích v rámci autonomního systému. OSPFv3 je definován v [8].

## BGP-4 (Border Gateway Protocol)

BGP-4 je externí směrovací protokol, což znamená, že je využíván ke směrování na nejvyšší úrovni mezi jednotlivými autonomními systémy. Z důvodu rozsahu směrovacích tabulek tyto okolním směrovačům nezasílá celé, pouze propaguje nastalé změny v topologii. Metrikou pro určení optimální cesty v tomto případě není pouze vzdálenost, ale do výpočtu je zahrnuto také množství dalších parametrů jako například politiky sítě, nebo pravidla nastavená administrátorem sítě. BGP je definován v [15], rozšíření pro IPv6 potom v [7].

### 2.2.3 Směrovací tabulka

Směrovací tabulka je datová struktura typicky přítomná v operační paměti směrovače. Tato uchovává agregovaná data o topologii sítě, podle kterých dokáže software směrovače rozhodnout, jakým způsobem zpracovat procházející datagramy. V tabulce se mohou nacházet dva druhy záznamu. Statický, zadaný administrátorem sítě, či dynamický, vytvořený během implementace některého směrovacího protokolu. V každé směrovací tabulce se musí nacházet několik základních informací, jsou to:

- **Cílová adresa** — IP adresa podsítě, nebo adresa konkrétní cílové stanice.
- **Síťová maska** — Filtr pro určení rozsahu IP adres, pro které je aktuální záznam použitelný.
- **Brána** — IP adresa nejbližšího směrovače na který je odeslán datagram v případě shody s aktuálním záznamem.
- **Rozhraní** — Identifikace síťového rozhraní na které je odeslán datagram v případě shody s aktuálním záznamem.
- **Metrika** — Priorita s jakou je záznam vybrán ve chvíli nalezení několika různých použitelných záznamů. Vlastní hodnota záleží na použitém směrovacím protokolu, případně zásahu administrátora.

Záznamy jsou ve směrovací tabulce seřazeny podle počtu jedničkových bitů síťové masky zleva sestupně. Začátek tabulky je tudíž zaplněn nejkonkrétnějšími záznamy, naproti tomu na konci tabulky jsou uvedeny záznamy nejobecnější. Toto uspořádání umožňuje optimálnější správu výjimek obecnějších pravidel.

Porovnání cílové adresy příchozího datagramu pak probíhá právě oproti síťové masce logickou funkcí **and**. Výsledný počet totožných bitů rovnající se po porovnání délce síťové masky vede na přijetí záznamu mezi vhodné kandidáty pro směrování datagramu.

## Kapitola 3

# Přidělování IPv6 adres

V této kapitole jsou popsány základní pojmy spojené s problematikou přidělování IPv6 adres a cíle, které přidělování sleduje. Dále je uvedena hierarchie internetových registrátorů a přehled pravidel, podle kterých je adresový prostor přidělován.

Obsah kapitoly čerpá z popisu politiky přidělování IPv6 adres [5] organizace RIPE NCC (Réseaux IP Européens Network Coordination Centre), kde není uvedeno jinak. Politiky ostatních internetových registrátorů se v dílčích detailech v závislosti na místních potřebách liší, do takových podrobností však obsah nezachází.

### 3.1 Základní pojmy

Pro dokonalé pochopení následujícího textu kapitoly bude nutné správně rozumět následujícím pojmům.

- **IANA (Internet Assigned Numbers Authority)** — Celosvětová organizace dohlížející na přidělování IP adres, správu kořenových zón DNS (Domain Name System) a typů médií pro MIME (Multipurpose Internet Mail Extensions). Dříve řízená vládou Spojených států amerických, nyní spadající pod organizaci ICANN (Internet Corporation for Assigned Names and Numbers).
- **Internetový registrátor (IR)** — Organizace zodpovědná za distribuci adresového prostoru IP dalším IR, cílovým zákazníkům a za správu záznamů o registracích. IR jsou kategorizovány podle své primární funkce a příslušnosti k místní oblasti. Hierarchii IR je znázorněna na obrázku 3.2.
- **Regionální internetový registrátor (RIR)** — Organizace ustanovená místní komunitou a uznaná organizací IANA. Reprezentuje větší geografické celky, jak je vidět na obrázku 3.1. Primární funkcí je správa a distribuce veřejného adresového prostoru v rámci regionu.
- **Národní internetový registrátor (NIR)** — Organizace zodpovědná za přidělování adresového prostoru lokálním registrům na státní úrovni. Tyto registry jsou zastoupeny především v asijsko-pacifickém regionu.
- **Lokální internetový registrátor (LIR)** — Organizace zodpovědná primárně za přidělování adresového prostoru uživatelům síťových služeb, které poskytuje. Obecně se jedná o ISP (Internet Service Provider), kteří poskytují své služby koncovým uživatelům, případně dalším ISP.

- **Přidělování** — Proces distribuce adresového prostoru jednotlivým IR za účelem jeho následné redistribuce.
- **Přiřazování** — Proces přidělení adresového prostoru koncovému uživateli nebo ISP za účelem využití v rámci vlastní síťové topologie. Tento adresový prostor nelze dále distribuovat. Proces musí být IR zdokumentován.
- **Využití** — Výjádření počtu přidělených síťových prefixů cílovým zákazníkům vzhledem k měrné jednotce, jíž je délka síťového prefixu /56.
- **HD (Host-Density) poměr** — Způsob měření účinnosti přiřazování adresového prostoru dle [9] takto:

$$HD = \frac{\log(X)}{\log(Y)}$$

kde

- **X** značí počet přidělených objektů.
- **Y** značí maximální počet přidělitelných objektů.

Objekty v tomto případě představují počet IPv6 adres přiřazených z IPv6 prefixu konkrétní délky.

- **Koncový uživatel (EU)** — Entita mající obchodní či právní vztah s poskytovatelem služeb, který zahrnuje:
  - Přiřazování adresového prostoru
  - Poskytování konektivity
  - Správu směrovacích údajů pro prefix zahrnující přiřazení EU

## 3.2 Cíle řízení přidělování IPv6 adres

K IPv6 adresovému prostoru se přistupuje jako k veřejnému zdroji, jenž musí být z této podstaty spravován s opatrností vzhledem k dlouhodobé udržitelnosti rozvoje internetu. Zodpovědná správa zahrnuje vyvážení několika konkrétních cílů, mezi nimiž v jistých situacích nastává konflikt. Cíle důležité pro správu IPv6 adresového prostoru jsou následující:

- **Jedinečnost** — Každému přidělení nebo přiřazení adresového prostoru musí být garantována celosvětová jedinečnost. Jedná se o absolutní požadavek na jednoznačnou identifikovatelnost veřejného hostitele v rámci celého internetu.
- **Registrace** — Přidělený adresový prostor musí být registrován v databázi přístupné pověřeným členům internetové komunity. Tento požadavek podporuje první cíl vyžadující jedinečnost každé adresy. Také poskytuje možnost odstraňování vzniklých problémů napříč celou hierarchií přidělování. Samotná registrace by měla splňovat rozumné zásady ochrany osobních údajů registrovaných entit.
- **Agregace** — Adresový prostor by měl být distribuován hierarchicky v závislosti na topologii sítě. Jedná se o prevenci expanze směrovacích tabulek a následných komplikací způsobených jejich velikostí. Vzhledem k velikosti adresového prostoru IPv6 je tento požadavek značně důležitý.

- **Konzervativnost** — Veškerá pravidla přidělování by měla být koncipována s ohledem na co nejmenší plýtvání adresovým prostorem i přes jeho značnou velikost. Každý požadavek na přidělení adresového prostoru by měl být například obhájěn adekvátní dokumentací.
- **Spravedlnost** — Všechna pravidla a procesy, vztahující se k využití adresového prostoru, by měly být aplikovány na každou entitu v procesu shodně, nezávisle na její lokaci, národnosti, velikosti atd.
- **Minimalizace režii** — Každý úkon spojený s přidělením adresového prostoru by měl být optimalizován z pohledu vlastního procesu i velikosti přidělovaného adresového prostoru, aby nedocházelo například ke zbytečné potřebě žádat o nové přidělení v rámci krátkého časového úseku apod.

Popsané cíle spolu ve spoustě případů mohou kolidovat, stejně tak jako s mohou přijít do konfliktu s požadavky konkrétních EU. Povinností každého IR je proto adekvátně vyvážit všechny vstupní požadavky a ospravedlnit zvolený individuální přístup řešení konfliktů. Přičemž je za nejdůležitější cíl považován požadavek na **Agregaci**.

### 3.3 Hierarchie internetových registrátorů

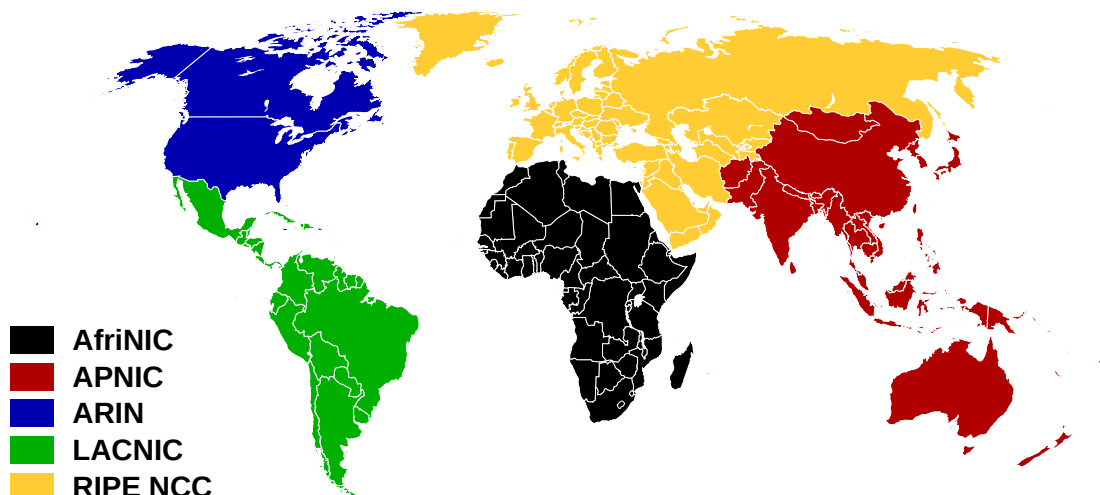
Celosvětově leží zodpovědnost za přidělování IPv6 adresového prostoru na organizaci IANA, jenž je řízena neziskovou organizací ICANN. Níže v hierarchii se nachází jednotlivý RIR, kterých aktuálně existuje pět, jde o tyto (Rozložení regionů příslušných RIR je znázorněno na mapě regionálních registrátorů, viz 3.1):

- **AFRINIC (African Network Information Center)** — Africký region
- **ARIN (American Registry for Internet Numbers)** — Spojené státy americké, Kanada a části karibské oblasti
- **APNIC (Asia-Pacific Network Information Centre)** — Asie, Austrálie, Nový Zéland a okolní oblasti
- **LACNIC (Latin America and Caribbean Network Information Centre)** — Latinská Amerika a části karibské oblasti
- **RIPE NCC (Réseaux IP Européens Network Coordination Centre)** — Evropa, Rusko, oblast Blízkého východu a centrální Asie

Dále je v hierarchii zařazena úroveň NIR. Tuto úroveň je možné nalézt zejména v regionu spravovaném organizací APNIC. Níže pak se v hierarchii nachází úroveň LIR. Jak již bylo zmíněno, jedná se převážně o větší ISP. Spodní úroveň potom tvoří koncový uživatelé – EU. Grafická reprezentace popsané hierarchie je znázorněna v grafu, viz 3.2.

### 3.4 Pravidla přidělování IPv6 adres

V této podkapitole jsou popsána zásadní pravidla přidělování adresových bloků mezi jednotlivými úrovněmi (viz 3.2) IR.



Obrázek 3.1: Mapa regionálních registrátorů (zdroj: Wikipedia [6])

### 3.4.1 IANA

Tento úsek textu popisuje pravidla přidělování IPv6 adresového prostoru organizací IANA regionálním internetovým registrátorům – RIR – dle [2].

- **Principy**

- Minimální velikost adresového bloku je stanovena na /12.
- IANA zaručí přidělení dostatečně velkého adresového prostoru, který pokryje požadavky RIR alespoň v následujících 18 měsících.
- IANA povolí RIR uplatňování vlastních pravidel přidělování, která zajistí splnění výše uvedených cílů, viz 3.2.

- **Počáteční přidělení**

- Každý existující RIR s přiděleným adresovým blokem menším než /12 obdrží alespoň tento.
- Každému nově vzniklému RIR bude přidělen adresový blok.

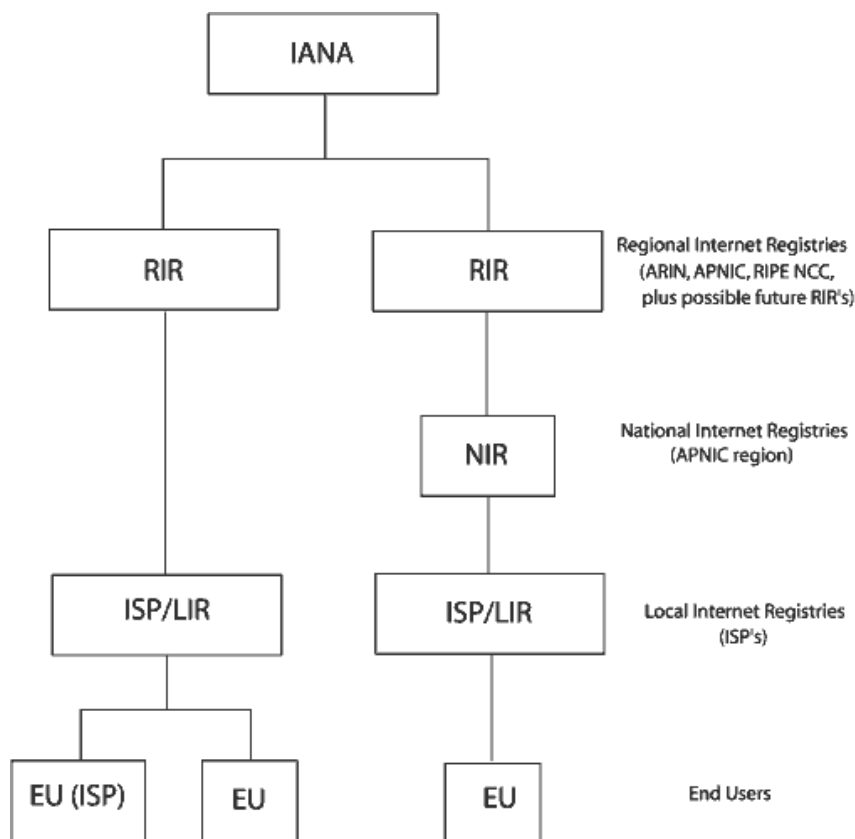
- **Dodatečné přidělení** — O dotatečné přidělení může RIR zažádat a bude mu vyhověno, pokud má RIR méně volného adresového prostoru než:

- 50% z bloku o velikosti /12
- Je velikost adresového prostoru, který se chystá RIR přidělit v následujících 9 měsících.

### 3.4.2 RIR

Tento úsek textu popisuje pravidla přidělování IPv6 adresového prostoru RIR lokálním internetovým registrátorům – LIR (případně NIR).

- **Počáteční podmínky** — Aby mohla organizace zažádat o přidělení adresového bloku, musí:



Obrázek 3.2: Hierarchie přidělování IPv6 adres (zdroj: RIPE NCC [5])

- Být LIR
- Mít vytvořený plán pro přidělování adresového prostoru dalším organizacím nebo EU v průběhu následujících 2 let.
- **Počáteční velikost adresového bloku** — Každá organizace splňující počáteční podmínky je hodná získání adresového bloku minimální velikosti /32. Pro přidělení většího počátečního bloku do velikosti /29 není třeba žádné dodatečné dokumentace. Ta je vyžadována s rozumným vysvětlením až pro požadavky na obdržení bloku většího než /29.
- **Dodatečné přidělení** – Organizace, která již má přidělený IPv6 adresový prostor může zažádat o dodatečné přidělení, pokud:
  - Splní podmínku hodnoty HD poměru [9] získaným z velikosti prefixu /56. Hodnota HD poměru byla stanovena na 0.94 [5].
  - V dokumentaci rozumným způsobem obhájí svůj požadavek.

Dodatečně přidělený adresový prostor by měl být stejně velký, jako již přidělený prostor. V ideálním případě bude přidělení provedeno z přiléhajícího adresového bloku, tak aby nedocházelo ke zbytečné fragmentaci adresového prostoru.

### 3.4.3 LIR/NIR

Neexistují žádná konkrétní pravidla přidělování IPv6 adresového prostoru LIR/NIR menším organizacím typu ISP, případně koncovým uživatelům – EU. Každý LIR si v kontextu cílů (3.2) může vyvinout pravidla vlastní. Existuje však soubor doporučení [14] IETF (Internet Engineering Task Force), podle kterých mohou LIR postupovat.

Závazné je pouze, že minimální velikost bloku přidělovaného EU je stanovena na /64 a každé přidělení adresového bloku /48 pro EU musí být LIR či ISP patřičně zaregistrováno, aby bylo možné na úrovni RIR určit HD poměr [9] v případě dodatečného přidělování adresového prostoru. Stejně tak platí, že adresové bloky větší než /48 budou přidělovány pouze s dostatečným zdůvodněním.

# Kapitola 4

## Současný stav

Komunikace v internetu je v současné chvíli postavena především na IP verze 4. Adresový prostor IPv4 však nebyl navržen pro tak rapidní nárůst počtu připojených zařízení, jaký se odehrál v poslední letech, a postupně se zmenšuje na hranici úplného vyčerpání. Tento fakt zapříčinil pozvolný přechod k IP verze 6. Vzájemná nekompatibilita těchto dvou verzí IP s sebou přináší potřebu vytvořit a optimalizovat nové vyhledávací algoritmy, které by mohly být nasazeny na směrovačích pro vyhledávání záznamů ve směrovacích tabulkách. Pro vytvoření takových algoritmů je potřeba mít dostatečné množství testovacích dat v podobě IPv6 prefixů odpovídajících realitě, ve které budou později – v budoucnu – tyto algoritmy pracovat.

Ranná fáze nasazení IPv6 technologie (Dle aktuálních statistik společnosti Google [3] pouze cca 23% síťového provozu probíhá na bázi IPv6) však neposkytuje možnosti jak získat reálná data vypovídající hodnoty pouhým extrahováním ze stávajících směrovacích tabulek. Problém spočívá v množství aktuálních záznamů, jenž má sice jistou vypovídající hodnotu, ale rozložením rozhodně neodpovídá budoucímu stavu. Je to způsobeno zejména politikou přidělování adresového prostoru s požadavkem na co nejefektivnější využití adresového prostoru, viz 3.2.

Východisko nastalé situace spočívá ve vytvoření automatického generátoru, který na základě různých faktorů vytvoří výsledek obsahující sadu prefixů s předpokládaným rozložením hodnot, tak jak by mohlo vypadat reálné rozložení hodnot prefixů v budoucnosti.

### 4.1 Přehled přístupů

Fází předcházející návrhu a implementaci vlastního generátoru IPv6 prefixů bylo nastudování existujících přístupů k vytvoření generátorů, které přinesly nové poznatky o možnostech, jakými uspokojit požadavek na vytvoření testovací sady prefixů. Z možných přístupů je třeba zmínit především následující dva:

#### 4.1.1 Non-random generator for IPv6 tables

Autoři generátoru [16] vychází z předpokladu, že IPv6 směrovací tabulky budou sdílet některé společné charakteristiky s aktuálními IPv4 tabulkami. Za cíl je tedy považováno generování takových IPv6 tabulek prefixů, které by tyto charakteristiky přítomné IPv4 tabulek převzaly a zároveň splňovaly požadavky IR na přidělování IPv6 adres.

V úvahu jsou brány tři hlavní aspekty. Hodnota prefixu, délka prefixu a velikost tabulky. Proces generování se zakládá na dodaném vstupním souboru, jenž obsahuje IPv4

prefixy z existující směrovací tabulky. K získání hodnoty IPv6 prefixu se přistupuje dvěma následujícími způsoby:

- **Vkládání bitů** — Jednotlivé postupy získání hodnoty prefixu z této skupiny pracují na principu vkládání bitů do existujícího IPv4 prefixu. V závislosti na pozicích, na které jsou bity vkládány, a samotném počtu bitů lze dosáhnout i odlišné úspěšnosti generování výstupu.
- **Přidávání čísla AS** — Postup přidávání čísla AS (Autonomous System), což je identifikátor označující část sítě pod jednotnou správou a využívající totožnou politiku směrování, je o něco náročnější. Ke každému IPv4 prefixu musí být nalezeno 16 bitové číslo AS, které je jako řetězcový prefix sloučeno s 32 bitovým identifikátorem IPv4 adresy. Náhodnými bity je výsledné číslo doplněno zprava do maximální délky /64 IPv6 prefixu.

Získání délky nového IPv6 prefixu je realizováno ořezáním hodnoty prefixu dvojnásobnou délkou délky původního IPv4 prefixu. Každá čtvrtá délka prefixu je s ohledem na distribuci prefixů upravena na lichý počet bitů.

Možností použít různé varianty získání hodnoty prefixu se generátor stává poměrně variabilním nástrojem. [16]

#### 4.1.2 V6Gene

Celkově odlišný přístup zvolili autoři generátoru V6Gene [17]. Jejich přístup vychází z již existující sady IPv6 prefixů. Nad kterou je prováděna simulace přidělování IPv6 adres podle pravidel IR (3).

Toto řešení vznikalo ve čase, kdy jednotliví LIR teprve žádali o přidělení adresového prostoru. Většina záznamů ve směrovacích tabulkách tak měla délku pouze /32. Zároveň záznamů neexistovalo mnoho. Proto autoři generátor doplnili i o část, která vytváří IPv6 prefixy náhodně.

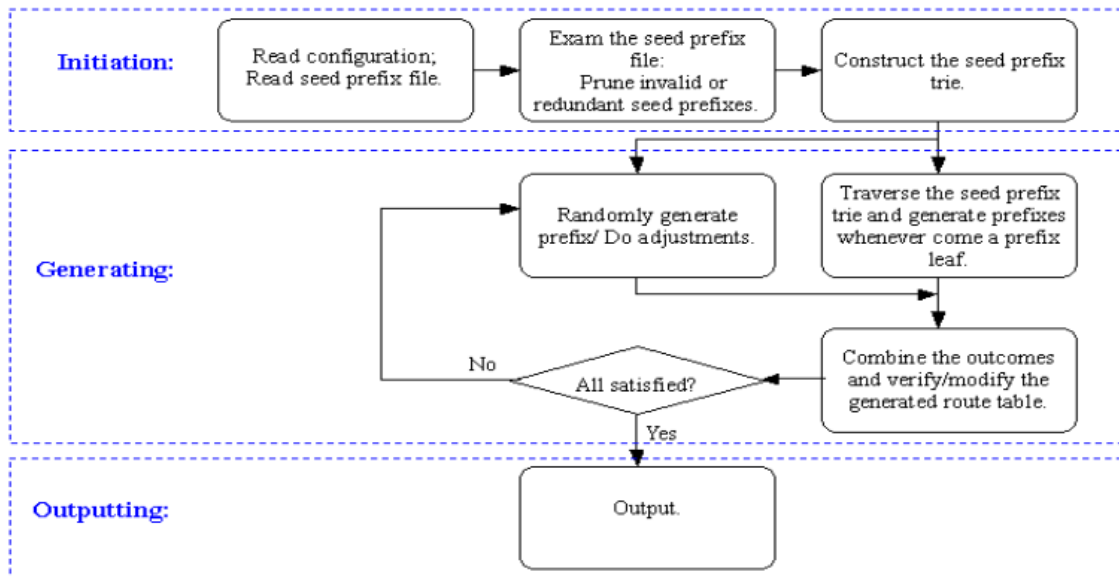
Celý proces generování se skládá ze tří fází (viz obrázek 4.1):

- **Inicializace** — V inicializační fázi je načten vstupní soubor s IPv6 prefixy a konfigurace v podobě parametrů. Dále jsou analyzovány prefixy ze vstupního souboru, přičemž jsou odstraněny všechny duplicitní či nevyhovující. Z takto ošetřené prefixové sady je sestaven binární prefixový strom.
- **Generování** — Fáze generování simuluje přidělování IPv6 adresových bloků a je rozdělena na dva paralelní procesy:
  - V prvním procesu probíhá procházení binárního prefixového stromu a při každém průchodu listovým uzlem je spuštěno generování, které vytvoří zadaný počet prefixů nové délky s požadovaným rozložením v závislosti na vstupních parametrech. Toto chování je simulací přidělování adresových bloků od existujících LIR směrem k EU.
  - Ve druhém procesu probíhá generování náhodně. Nejprve je vytvořen náhodný počet prefixů s délkou korespondující délce prefixů LIR, což simuluje vytváření nových LIR, nebo těch, kteří si ještě nezažádali o přidělení adresového bloku. Z těchto jsou potom vytvořeny na základě parametrů nové prefixy simulující opět přidělování prefixů od LIR směrem k EU.

Po skončení obou procesů je výsledek sloučen a zkontrolován na výskyt duplicit, které jsou odstraněny. Odpovídá-li výsledek zadaným parametrům, proces generování se přesouvá do výstupní fáze. V opačném případě se celý proces opakuje do chvíle, kdy jsou splněny veškeré požadavky.

- **Výstup** — Ve výstupní fázi jsou všechny vygenerované prefixy zformátovány do podoby vstupního souboru a předány na výstup.

Z popisu autorů vyplývá, že hlavní benefit, který implementace tohoto generátoru přináší, spočívá v podobě rozsáhlého množství nastavení, jenž mohou různými způsoby ovlivnit skladbu výstupních dat. Na škodu je pouze fakt, že původní implementace autorů není přístupná pro porovnání výsledků generování. [17]



Obrázek 4.1: Průběh generování V6Gene (zdroj: V6Gene [17])

## Kapitola 5

# Generátor IPv6 prefixů dle V6gene

Návrhu a implementaci vlastního generátoru předcházela implementace přístupu generování IPv6 prefixů popsaná v článku [17]. Od prvního představení přístupu V6gene uběhlo několik let. V průběhu času tak došlo k přidělení více adresových bloků. To obnáší i růst počtu zařízení využívající technologii IPv6. Je tedy možné získat i, vzhledem k budoucímu rozložení prefixů, relevantnější vzorky dat ze směrovacích tabulek. Jelikož princip V6gene vychází z existujících sad IPv6 prefixů a současně poskytuje široké možnosti parametrizace, existuje úvaha, že by mohl na základě aktuálních dat tento přístup vytvářet uspokojivé výsledky. Práci lze navíc využít jako základ pro pozdější vlastní implementaci generátoru.

### 5.1 Návrh

Na začátku inicializační fáze je nutné načíst parametry konfigurace a vstupní soubor s IPv6 prefixy. Pomocí parametrů by mělo být možné zadat požadovaný počet prefixů na výstupu, poměr RGR (Random Generating Ratio) tak jak je představen v [17] pro určení velikosti části výsledných prefixů, které budou generovány nezávisle na vstupním souboru. Dále pak parametr ve kterém bude možné zadat seznam požadovaných délek prefixu, jenž se mají objevit na výstupu. Také bude třeba předat v parametrech požadavek na zanoření prefixů v rámci cesty prefixového stromu – tzv. level.

Kontrola vstupního souboru s IPv6 prefixy bude probíhat na základě předpokladu, že neexistuje žádný prefix v rozsahu délek od /65 do /127 [17]. Také budou odstraněny veškeré nalezené duplicitní prefixy. Z takto ošetřeného vstupu bude sestaven binární prefixový strom.

Uzel prefixového stromu představující prefix ponese informaci o úrovni. Ta bude sloužit pro kontrolu požadavků na výslednou distribuci prefixů na základě úrovně.

Ve fázi generování bude nutné vytvořit mechanismus pro vytvoření skupiny náhodných bitů. Délka této skupiny bitů bude odpovídat rozdílu délek prefixu ze vstupního souboru a požadované délky prefixu nového. V případě nezadaného parametru s délkami výsledných prefixů budou k dispozici výchozí délky generovaných prefixů dle 3. Tedy bude nutné nastavit práh délek pro každou úroveň prefixů (RIR, LIR, EU). Na základě takto definovaných intervalů bude možné podle délky vstupního prefixu rozhodnout, jakou délku přiřadit generovanému prefixu.

Ověření naplnění požadavků generování bude obsahovat kromě logiky kombinující výstupy generování také mechanismus kontroly uvážnutí generátoru. Část generující výstupy na základě vstupního souboru se ve spojení s parametrem na rozložení délek prefixů může

dostat do stavu, kdy bude v extrémním případě požadováno vytvoření většího množství prefixů, než bude možné vytvořit z bitového rozsahu rozdílů požadovaných délek prefixů a délek prefixů vstupního souboru. Dojde-li tedy k uvážnutí generátoru bude adaptována technika popsaná v [17], kdy převezme splnění požadavků na vytvoření prefixů náhodný generátor.

Výstupní fáze pouze zajistí předání výsledku na výstup. Volitelně bude umožněno na výstupu spojit nově vytvořené prefixy s těmi ze vstupního souboru.

## 5.2 Implementace

Generátor byl implementován jako konzolová aplikace v jazyce Python 3 s následujícími parametry.

### 5.2.1 Parametry

Jednotlivé parametry nejsou povinné. Bez vstupního souboru se spouští pouze náhodný generátor. Bez zadání výstupního souboru se tiskne výsledek na standardní výstup.

- **-i, -input=**  
Nastaví vstupní soubor s prefixy.
- **-o, -output=**  
Nastaví výstupní soubor pro generované prefixy.
- **-p, -prefixes=**  
Nastaví požadovaný počet prefixů na výstupu. Počet musí být zadán jako celé číslo.
- **-l, -level=**  
Nastaví maximální povolenou úroveň prefixu. Úroveň musí být zadán jako celé číslo.
- **-rgr=**  
Nastaví poměr počtu prefixů generovaných nezávisle na vstupním souboru ( $R$ ) k celkovému počtu generovaných prefixů ( $N$ ). Poměr může být zadán jako celé nebo desetinné číslo vyjadřující procentuální část náhodně generovaných prefixů z celku dle vzorce:

$$\frac{R}{N} * 100$$

- **-scatter=**  
Nastaví seznam délek prefixů požadovaných na výstupu. Seznam délek tvoří celá čísla oddělená znakem ','. Validní délky prefixů jsou v rozsahu intervalu  $\langle 0, 64 \rangle$ . Prefixy na výstupu budou mít pouze uvedené délky. Nelze garantovat, že se všechny uvedené délky na výstupu objeví, což závisí především na obsahu vstupního souboru.
- **-keep-seeds**  
Zachová validní prefixy ze vstupního souboru na výstupu.
- **-help**  
Zobrazí nápovědu k aplikaci.
- **-version**  
Zobrazí informace o verzi.

### 5.2.2 Inicializace

V první řadě jsou načteny parametry (5.2.1). Vyskytne-li se zásadní nedostatek např. hodnota parametru mimo logický rozsah, je reportována chyba.

Následně je zpracováván vstupní soubor. Předpokládané formátování souboru je jeden prefix na samostatném řádku. Každý prefix je uložen jako objekt třídy `tt ipaddress.IPv6Network` do slovníku prefixů, což zajistí splnění požadavku na jedinečnost a validitu prefixů vstupní sady. Také je prováděna filtrace s parametrem délky prefixu a typu prefixu s omezením pouze na Globální unicast (2.1).

Z načtených prefixů je vytvořen binární prefixový strom. Úroveň prefixu je definována způsobem, kdy je třeba při vložení dalšího prefixu do stromu tuto hodnotu přepočítat pro všechny prefixy minimálně na cestě, na kterou byl prefix uložen. Proto byl zvolen postup implementace inverzní hodnoty úrovně prefixu, kterou lze počítat již při vkládání prefixu do struktury stromu, přičemž následné kontrole úrovní generovaných prefixů naprosto dostačuje.

### 5.2.3 Generování

Generátor byl vytvořen s hierarchickou strukturou, kde společné funkce pro manipulaci s binárním řetězcem a vytvářením nových prefixů poskytuje třída obecného generátoru. Specifické varianty generátoru s drobnými odlišnostmi v logice fungování jsou vytvořeny zvlášť.

Intervaly pro určení délky prefixu byly nastaveny dle pravidel pro přidělování adresového prostoru (3) takto:

- **RIR** /12 až /23
- **LIR** /23 až /32
- **EU** /33 až /64

Výchozí délky prefixů byly nastaveny dle [17] a 3 na:

- **RIR** /12
- **LIR** /32
- **ISP** /48
- **EU** /64

Ověření uváznutí se kontroluje podle počtu vygenerovaných prefixů v rámci jednoho průběhu generátoru prefixovým stromem. Není-li vygenerován ani jeden nový prefix, přičemž stále existuje požadavek na určitý počet vygenerovaných prefixů, je automaticky předán zbývající počet prefixů generátoru čistě náhodných prefixů.

Filtrace duplicit prefixů vytvořených ze vstupního souboru a náhodně probíhá automaticky díky jednotnému uložení generovaných prefixů a vhodně zvolenému datovému typu `set`.

Vlastní proces generování probíhá ve shodném sledu pro oba dva typy generátoru (rozdíly jsou patrné pouze v parametrech volání) následovně:

1. Určení délky nového prefixu na základě délky stávajícího (listový prefix různé délky či náhodně generovaný prefix délky v rozsahu LIR)
2. Získání počtu náhodně generovaných bitů z rozdílu délek nového a stávajícího prefixu
3. Generování náhodného čísla a konkatenace bitů (Náhodné bity jsou s k původnímu prefixu přidány zprava.)
4. Převedení binární reprezentace do hexadecimální
5. Uložení výsledku

### 5.2.4 Výstup

Ve výstupní fázi se pouze tiskne výsledek generování do určeného souboru, nebo na standardní výstup v případě, že soubor nebyl zadán.

## 5.3 Spuštění

Z důvodu nedostupné referenční implementace nelze přesně ověřit, zda-li vytvořený generátor přistupuje k řešení problematiky některých sekcí článku [17], které stylem svého popisu dovolují velmi volný přístup k návrhu, totožně, jak bylo původně zamýšleno. V této situaci lze k ověření přistoupit například následujícím způsobem.

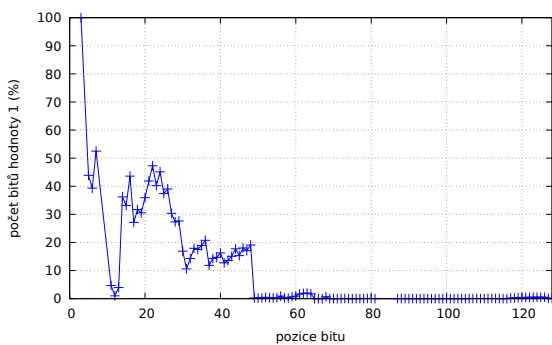
### 5.3.1 Vstupní data a parametrizace

Za předpokladu že máme přístup k historickým i aktuálním datům směrovacích tabulek [1], je možné vybrat určitý vzorek historických dat, z něj odvodit závislosti, které lze předat v podobě parametru generátoru, a s těmito parametry spustit generování. Nově vygenerovaná data je poté možné porovnat s aktuální verzí dat z databáze, na základě čehož si můžeme dovojit vyslovení závěru o schopnostech implementace.

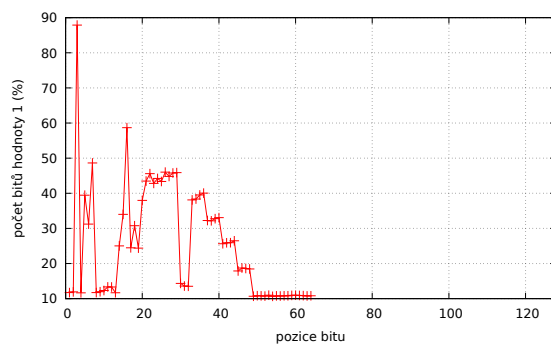
Pro účel spuštění byla tedy vybrána data z archivu [1], pro tento účel konkrétně obsah adresáře `ipv6/route-views.eqix/bgpdata` (Použité nástroje pro filtraci a postup získání dat viz následující kapitola 6). Jednotlivé vzorky byly vybrány s časovým rozestupem čtyřech let, konkrétně roky: 2010, 2014 a 2018.

Provedením analýzy dat (viz kapitola 6) vzorku z roku a 2014 bylo zjištěno rozložení počtu prefixů v závislosti na jeho délce. Z tohoto rozložení byly pro parametr požadovaného rozložení délek (`scatter`) vybrány ty, jenž se vyskytovaly ve vzorku minimálně v 1% případech. Parametr udávající počet požadovaných prefixů na výstupu byl nastaven na hodnotu rozdílu počtu prefixů z roku 2018 a 2010, přičemž bylo nastaveno požadované sloučení prefixů vstupního souboru a nově generovaných prefixů. Počet náhodně vygenerovaných prefixů byl nastaven na hodnotu 15%. To mělo za následek lehce patrné zvětšení počtu prefixů ve výstupní sadě oproti vzorku z roku 2018.

Celý experiment (`test_v1_future_2`) je možné detailně prostudovat z příložených souborů A, jelikož bylo jeho spuštění implementováno jako automatická úloha.

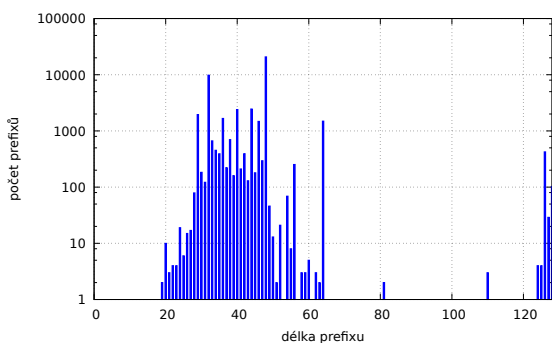


(a) Routeviews (Equinix Ashburn) 1.1.2018

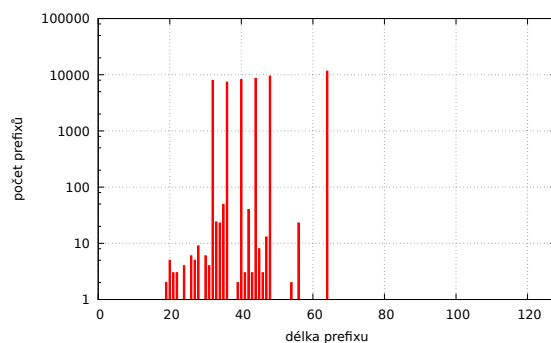


(b) Výstup generátoru

Obrázek 5.1: Porovnání rozložení bitů s hodnotou 1 v reálném vzorku z roku 2018 a výstupu implementovaného generátoru.

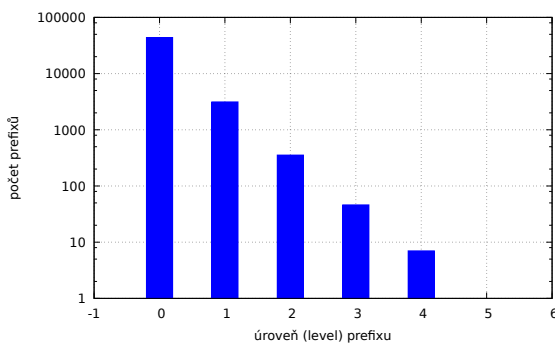


(a) Routeviews (Equinix Ashburn) 1.1.2018

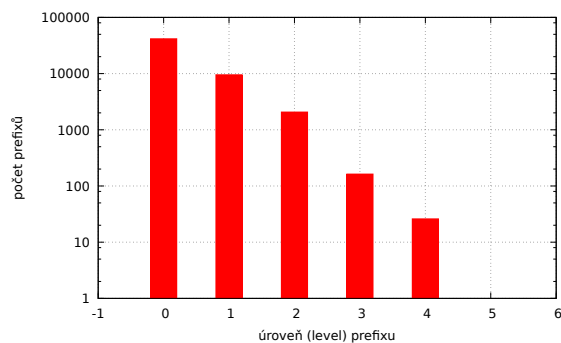


(b) Výstup generátoru

Obrázek 5.2: Porovnání rozložení délek prefixů v reálném vzorku z roku 2018 a ve výstupu implementovaného generátoru. (Povšimněte si logaritmického měřítka osy y)

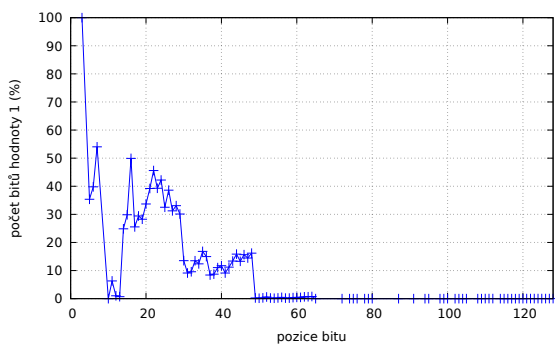


(a) Routeviews (Equinix Ashburn) 1.1.2018

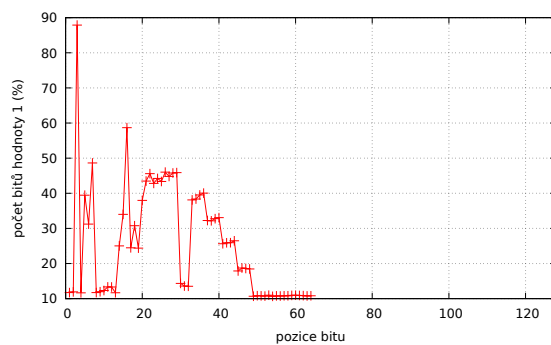


(b) Výstup generátoru

Obrázek 5.3: Porovnání rozložení úrovní prefixů v reálném vzorku z roku 2018 a ve výstupu implementovaného generátoru. (Povšimněte si logaritmického měřítka osy y)

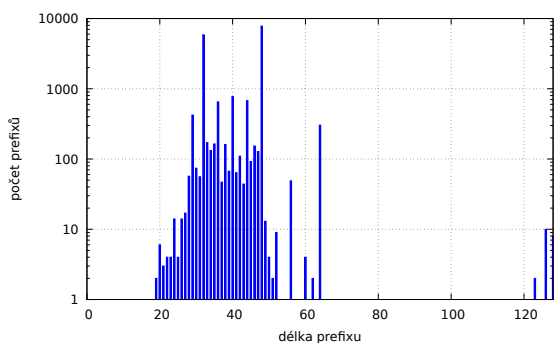


(a) Routeviews (Equinix Ashburn) 1.6.2014

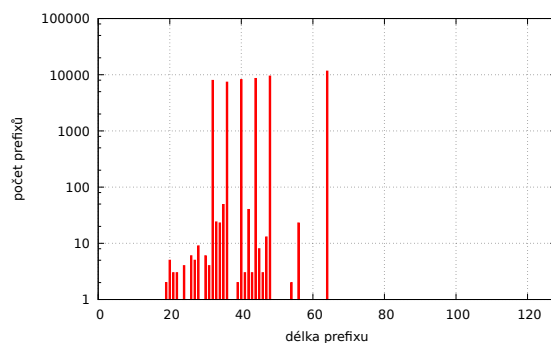


(b) Výstup generátoru

Obrázek 5.4: Porovnání rozložení bitů s hodnotou 1 v reálném vzorku z roku 2014 a výstupu implementovaného generátoru.

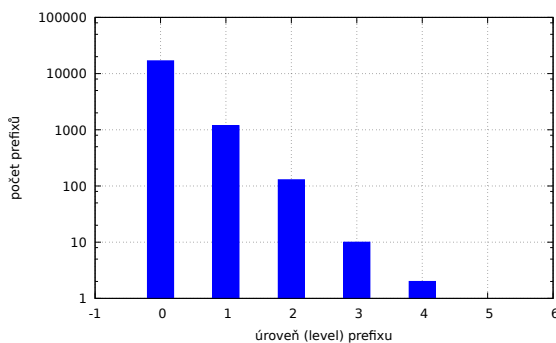


(a) Routeviews (Equinix Ashburn) 1.6.2014

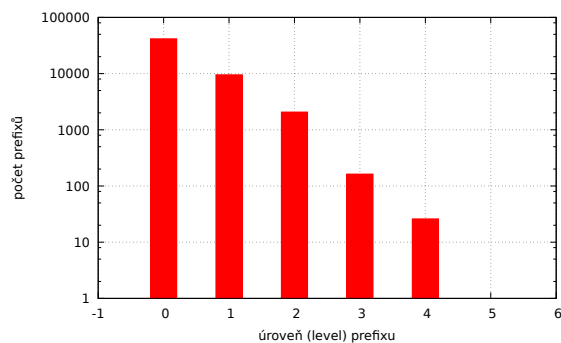


(b) Výstup generátoru

Obrázek 5.5: Porovnání rozložení délek prefixů v reálném vzorku z roku 2014 a ve výstupu implementovaného generátoru. (Povšimněte si logaritmického měřítka osy y)



(a) Routeviews (Equinix Ashburn) 1.6.2014



(b) Výstup generátoru

Obrázek 5.6: Porovnání rozložení úrovní prefixů v reálném vzorku z roku 2014 a ve výstupu implementovaného generátoru. (Povšimněte si logaritmického měřítka osy y)

## Kapitola 6

# Analýza reálných IPv6 prefixových sad

V této kapitole jsou popsány zdroje dat a metodiky analýzy aplikovatelné při zkoumání závislostí ve vývoji reálných prefixových sad, ale i prefixových sad nově generovaných. Dále je v kapitole uvedeno pozorování časového vývoje rozložení prefixů. Na závěr je představeno vyhodnocení výsledků a jeho srovnání s výstupy generování.

### 6.1 Zdroje reálných dat

K analýze bylo třeba získat vypovídající testovací data. Potřebné informace pro tento blok činnosti lze získat z několika možných zdrojů, jimiž jsou v některých případech projekty světových univerzit, otevřené databáze organizací, jež spravují některé síťové uzly, případně výsledky zájmové činnosti skupiny lidí či jednotlivců zabývajících se touto problematikou. Pro tyto účely byla data získána z archivu směrovacích tabulek Routeviews [1] a archivu organizace RIPE NCC [4]. Nejedná se o nikterak rozsáhlý vzorek, nicméně byl vybrán s důrazem na jeho variabilitu. Jednotlivé obsahy směrovacích tabulek pocházejí z lokalit datových center Spojených států amerických (Ashburn), Brazílie (São Paulo) a Evropy (Amsterdam). Tyto lokality jsou významné zejména proto, že v nich probíhá velké množství IPv6 komunikace (viz [3]) oproti ostatním místům.

### 6.2 Metodiky analýzy

Pohledů na rozbor dat je k dispozici hned několik. Prvním z nich může být závislost počtu prefixů na jejich délce, jak je popsáno v [17]. Jinými slovy kolik prefixů délky např. /48 se ve vzorku vyskytuje. Sledováním tohoto kritéria v čase mezi vzorky z jednoho zdroje je možné zjistit, jak fungují pravidla IR pro přidělování adresového prostoru nebo počet nově vzniklých uzlů v síti atp.

Další informace o charakteru dat může prozradit způsob analýzy osvětlený v článku [16]. Jedná se o pozorování rozložení samostatných bitů prefixu. Což znamená dívat se např. na první bit všech prefixů a počítat, kolik prefixů z celku má první bit roven hodnotě 1. Stejně pak pro ostatní pozice. Pomocí tohoto kritéria lze opět sledovat přímé závislosti s pravidly pro přidělování adresového prostoru, zejména potom požadavek na agregaci (viz 3.2).

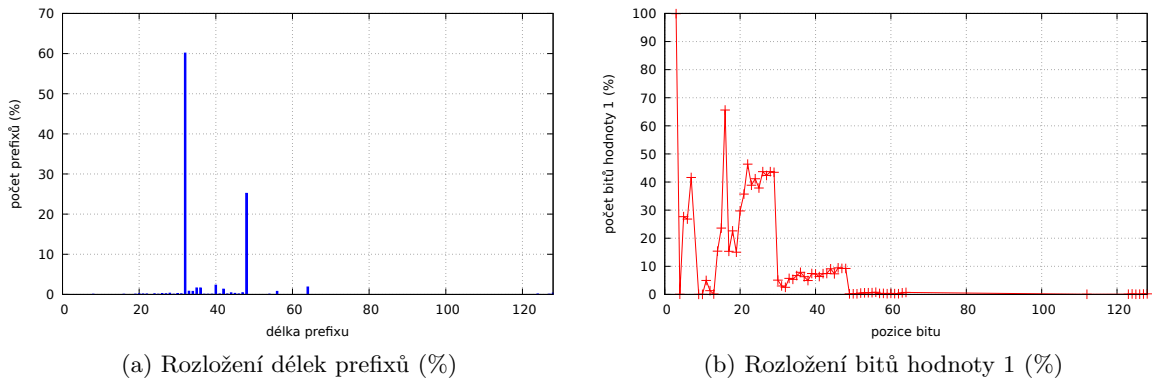
V neposlední řadě je možné sledovat vývoj struktury sítě pozorováním úrovně prefixů (viz [17]). Ta vyjadřuje informaci o hierarchii podsítě.

### 6.3 Provedení analýzy

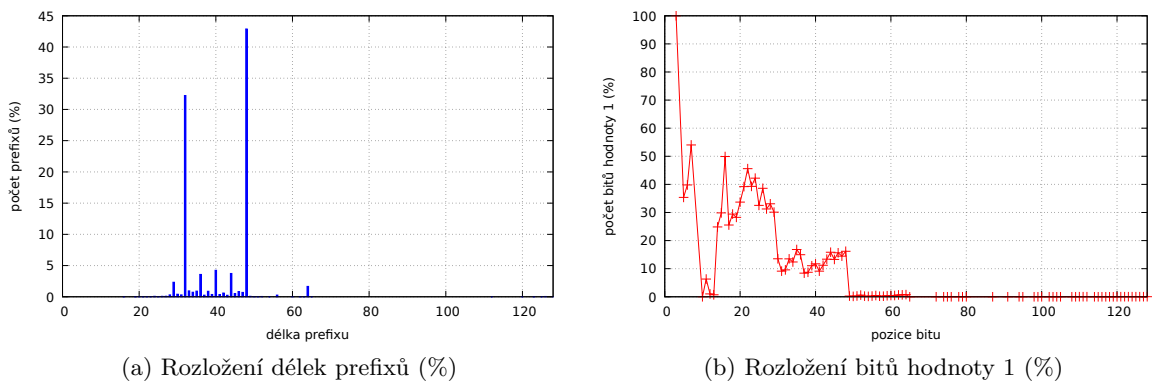
Zpracování vstupů pro analýzu proběhlo automaticky za pomoci sady skriptů vytvořené za tímto účelem. Jádrem sady tvoří skript napsaný v jazyce Python 3, jehož hlavní úlohou je načíst IPv6 prefixovou sadu a podle zadaných parametrů spočítat rozložení bitů v prefixech, rozložení délek prefixů v sadě, nebo úroveň každého prefixu. Ostatní skripty jsou shellové skripty zajišťující dávkové zpracování a vytváření grafického výstupu. Celou sadu skriptů je možné nalézt v příloze A.

### 6.4 Časový vývoj prefixové sady

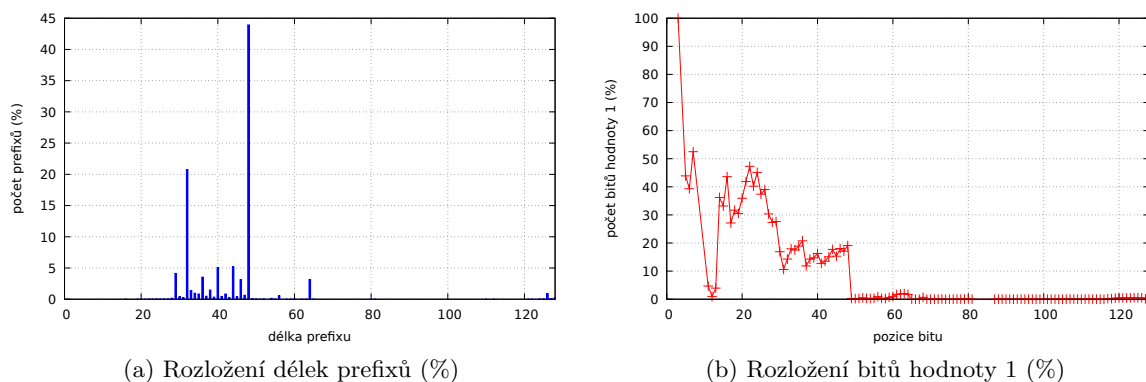
Pozorováním vývoje rozložení počtu prefixů v závislosti na jejich délce by bylo pravděpodobně možné zdůvodnit požadavek na větší škálovatelnost nastavení generátoru. Existuje totiž domněnka, že se nové prefixy, v závislosti na pravidlech IR a aktualizovaném doporučení [14], budou v blízké budoucnosti více kumulovat mezi délkami /40 a /56.



Obrázek 6.1: Stav prefixové sady v roce 2010 (Routeviews — Equinix Ashburn)



Obrázek 6.2: Stav prefixové sady v roce 2014 (Routeviews — Equinix Ashburn)

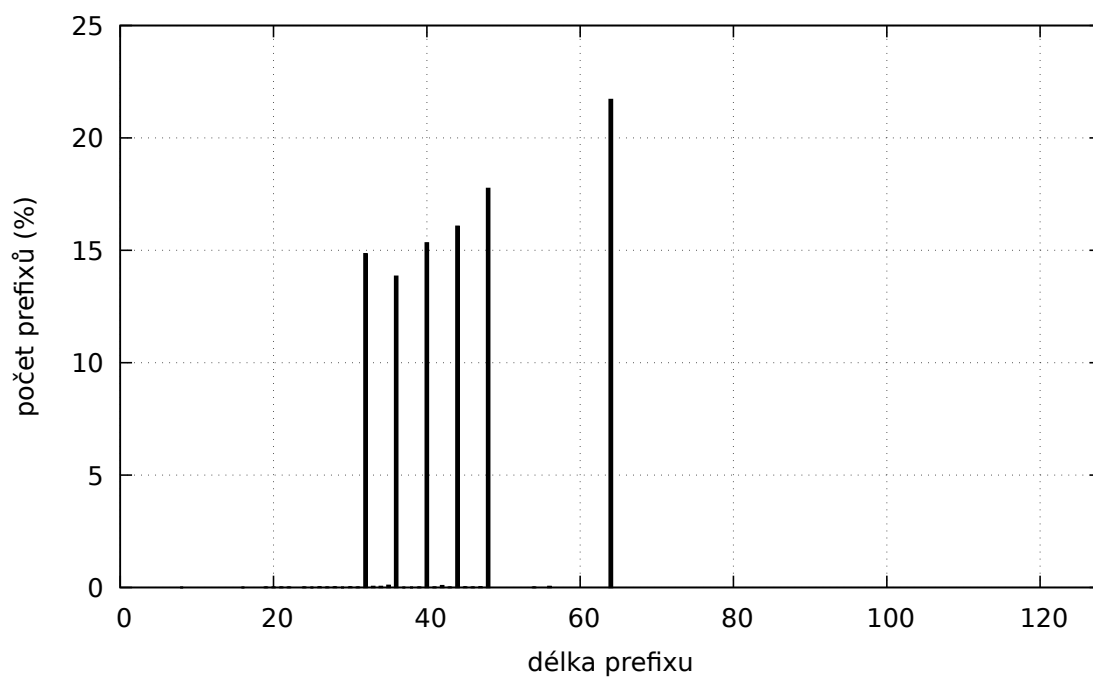


Obrázek 6.3: Stav prefixové sady v roce 2018 (Routeviews — Equinix Ashburn)

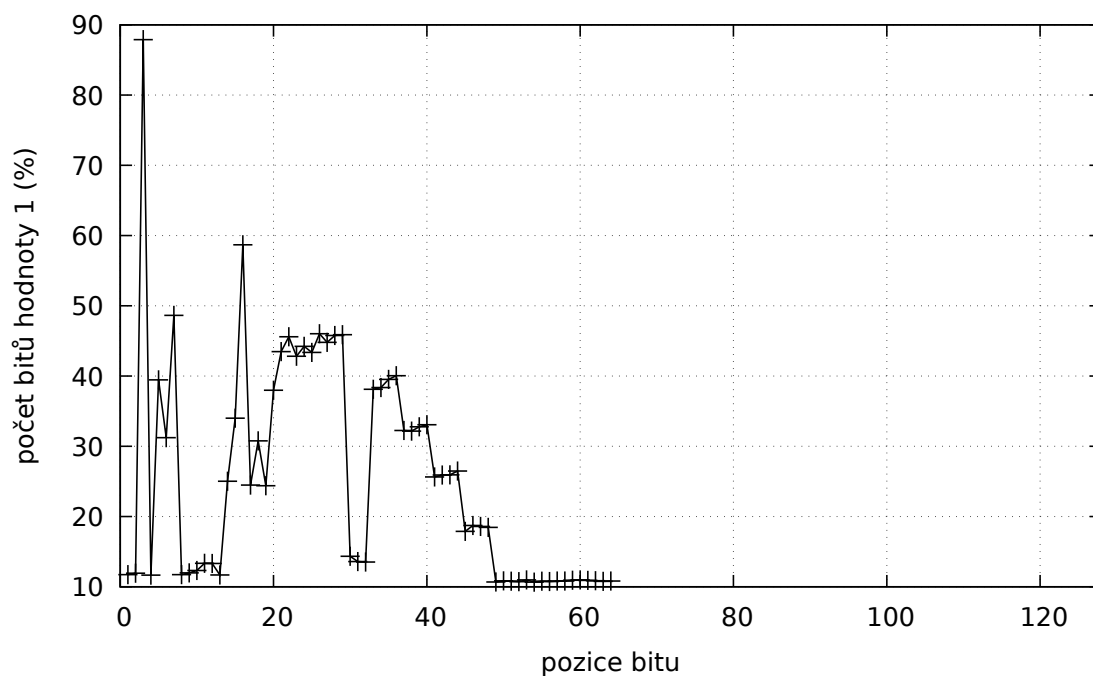
## 6.5 Vyhodnocení

Vývoj prefixové sady, která je znázorněna sérií zaznamenaných stavů 6.1, 6.2 a 6.3, ale i dalších pozorovaných sad vykazuje oproti původnímu očekávání zajímavý trend, kdy dochází k selektivnímu růstu zastoupení délek v rozmezí /32 až /48. Ten může být způsobený příchodem organizací, jenž si nárokují větší adresové bloky než výchozí /48. Udržení takového stavu však potrvá pouze omezeně dlouhou dobu, neboť těchto organizací nebude existovat mnoho.

I když se tedy zatím nepotvrdila myšlenka rychlejšího nárůstu prefixů větší délky, ze srovnání aktuálního stavu vývoje a 6.3 a výsledků generování 6.4 vyplývá, že aktuální implementace generátoru nedokáže dostatečně přesně reflektovat drobné rozdíly růstu počtu prefixů ve středech intervalů mezi minimálními přidělovanými délkami. Rozložení hodnoty bitů generované sady 6.5 je v porovnání částečně správné. Velké výkyvy hodnot jsou na spodní hranici délky způsobené především konzervativní politikou organizace IANA, která momentálně přiděluje pouze z několika málo adresových bloků, naproti tomu generátor vytváří náhodné prefixy bez vazby na tuto politiku. Markantní rozdíly v ostatních částech grafu jsou zapříčiněny právě nevhodným rozložením délek.



Obrázek 6.4: Rozložení délek prefixů generované sady (%)



Obrázek 6.5: Rozložení bitů hodnoty 1 (%) generované sady

## Kapitola 7

# Návrh a implementace vlastního generátoru

V této kapitole je popsán návrh vlastního generátoru IPv6 prefixů a jeho příslušná implementace. Ta vychází z prvotní realizace generátoru 5. Dále jsou v kapitole uvedeny výsledky testování tohoto generátoru.

### 7.1 Požadavky

Poždavkem na výsledek práce bylo navrhnout generátor IPv6 prefixů se znalostí aktuálních pravidel přidělování IPv6 adresového prostoru. Výstupem aplikace by tak měly být hodnoty prefixů co nejvíce podobné realitě.

Kapitola 6 ukázala, že původně vytvořený generátor 5 skrývá jistý potenciál na doručení kvalitnějšího výsledku při vhodnějším způsobu parametrizace, zejména potom v části upravující rozložení délek prefixů.

### 7.2 Návrh

Logická posloupnost dílčích částí celkového procesu bude z většiny zachována. Na začátku běhu aplikace bude konvenčně potřeba načíst vstupní parametry s nastavením procesu generování. Už tvar vstupních parametrů musí reflektovat požadavek na vylepšení. Dále bude nutné načíst a zpracovat vstupní soubor s referenční sadou prefixů. Načítání vstupního souboru lze provést téměř totožně. Fázi generování bude nutné přepracovat v závislosti na potřebě dodatečných kontrol a změny struktury ukládání generovaných prefixů. Výstupní fáze může v případě zachování výstupu fáze generování zůstat kompletně zachována.

#### 7.2.1 Parametry

Jak je uvedeno v kapitole 6, aby byl generátor schopen reflektovat nerovnoměrný přírůstek počtu prefixů v závislosti na jejich délce, musí být možné tuto skutečnost generátoru adekvátní formou sdělit. Řešení se nabízí v podobě rozšíření aktuálního parametru `scatter`. Ideálně by bylo vhodné zadávat s každou požadovanou délkou prefixu i hodnotu definující podíl výskytu této délky ve výsledné sadě prefixů, podobně jako je možné zadat parametrem `rgr` počet náhodně generovaných hodnot.

S ohledem na požadavky přidělování IPv6 adres [3.2](#), konkrétně agregaci adresového prostoru, bude vhodné přidat další možnost parametrizace udávající nastavení vytváření přílehlých bloků IPv6 adres z náhodně generovaných (viz dále).

## 7.2.2 Generování

Se skutečností existujícího požadavku na konkrétní rozložení délek prefixů ve výsledné sadě se pojí potřeba odlišného přístupu k vnitřní reprezentaci uložených hodnot. Ty by měly být ukládány zvlášť ve skupinách podle délky prefixu, aby byla realizovatelná kontrola výsledného počtu. Aby se předešlo dvojí kontrole, uložště vytvářených prefixů může slučovat výsledky generování ze vstupního souboru s výsledky náhodného generování.

Pro dodržení požadavku na agregaci adresového prostoru (viz [3.2](#)) připadá v úvahu postup, v rámci kterého se z každého nově vytvořeného prefixu odvodí několik následujících prefixů v číselné řadě. Tzn. všechny prefixy budou mít shodných prvních několik bitů. Kolik posledních bitů bude odlišných, zajistí nový parametr.

Nový generátor by měl při výchozím nastavení také odrážet změny ve výchozích délkách přidělovaných prefixů koncovým uživatelům (viz [\[14\]](#)).

## 7.3 Implementace

Generátor byl stejně jako v předchozí verzi (viz [5](#)) implementován jako konzolová aplikace v jazyce Python 3 s následujícími parametry.

### 7.3.1 Parametry

Parametry nejsou povinné. Chybějící vstupní soubor znamená spuštění náhodného generátoru. Bez zadání výstupního souboru se tiskne výsledek na standardní výstup. Výchozí hodnoty parametrů `derived-step` a `derived-max` jsou nastaveny na základě vyhodnocení testovacích výstupů.

- `-i, -input=`  
Nastaví vstupní soubor s prefixy.
- `-o, -output=`  
Nastaví výstupní soubor pro generované prefixy.
- `-p, -prefixes=`  
Nastaví požadovaný počet prefixů na výstupu. Počet musí být zadán jako celé číslo.
- `-l, -level=`  
Nastaví maximální povolenou úroveň prefixu. Úroveň musí být zadán jako celé číslo.
- `-rgr=`  
Nastaví poměr počtu prefixů generovaných nezávisle na vstupním souboru ( $R$ ) k celkovému počtu generovaných prefixů ( $N$ ). Poměr může být zadán jako celé nebo desetinné číslo vyjadřující procentuální část náhodně generovaných prefixů z celku dle vzorce:

$$\frac{R}{N} * 100$$

- **-scatter=**  
Nastaví seznam délek prefixů požadovaných na výstupu. Seznam délek tvoří buďto celá čísla oddělená znakem ',', nebo skupiny dvojic celé číslo (značící délku prefixu) – desetinné číslo (značící procentuální výskyt ve výsledné sadě), mezi kterými se nachází znak '-', jednotlivé skupiny jsou pak odděleny znakem ',.'. Tyto dva přístupy není možné kombinovat. Validní délky prefixů jsou v rozsahu intervalu  $< 0,64 >$ . Prefixy na výstupu budou mít pouze uvedené délky. Nelze garantovat, že se všechny uvedené délky na výstupu objeví, což závisí především na obsahu vstupního souboru. V případě zadání procentuálního rozložení budou délky v součtu přesahující hodnotu 100% ze seznamu odstraněny.
- **-derived-step=**  
Nastaví maximální krok mezi dvěma odvozenými prefixy. Krok musí být zadán jako celé číslo. Výchozí hodnota kroku je 1.
- **-derived-max=**  
Nastaví maximální počet odvozených prefixů pro každý generovaný prefix. Počet musí být zadán jako celé číslo. Výchozí hodnota je 10.
- **-keep-seeds**  
Zachová validní prefixy ze vstupního souboru na výstupu.
- **-help**  
Zobrazí nápovědu k aplikaci.
- **-version**  
Zobrazí informace o verzi.

### 7.3.2 Inicializace

Inicializační fáze generátoru zůstala oproti původnímu návrhu (viz 5) zachována v rozmezí načítání a validace vstupního souboru. Čtení konfigurace generátoru ze vstupních parametrů bylo upraveno do stavu vyhovujícího popisu parametrů, který je uveden v předcházejícím odstavci.

### 7.3.3 Generování

Generování prefixů je opět rozděleno do dvou částí. Kdy v první probíhá procházení binárního prefixového stromu vytvořeného v inicializační fázi. Kdykoliv je při procházení nalezen listový uzel, je spuštěn generační proces. Ve druhé části se spouští náhodné generování celé délky prefixu nezávisle na obsahu vstupního souboru.

Postup vytváření probíhá ve shodně pro oba dva typy generátoru (rozdíly jsou patrné opět pouze v parametrech volání) a zahrnuje tyto kroky:

1. Určení délky nového prefixu na základě délky stávajícího (listový prefix různé délky či náhodně generovaný prefix délky v rozsahu LIR)
2. Získání počtu náhodně generovaných bitů z rozdílu délek nového a stávajícího prefixu
3. Generování náhodného čísla a konkatenace bitů (Náhodné bity jsou s k původnímu prefixu přidány zprava.)

4. Převedení binární reprezentace do hexadecimální
5. Na základě hodnot parametrů `derived-step` a `derived-max` se spouští odvození následujících prefixů:
  - (a) Převedení hodnoty vygenerovaného prefixu na celé číslo v desítkové soustavě
  - (b) Přičtení velikosti kroku
  - (c) Validace odvozené hodnoty
  - (d) Převedení hodnoty do hexadecimální reprezentace prefixu
  - (e) Opakování postupu podle hodnoty `derived-max`
6. Uložení výsledku

Každý nově vytvořený prefix je ukládán v rámci jednoho generovacího cyklu do lokálního uložení objektu generátoru. Na konci jednoho průběhu generovacího cyklu jsou prefixy předány do správy globálnímu uložení, které bylo vytvořeno za účelem jednotné správy množství generovaných prefixů a kontroly požadavků na jejich rozložení. Prefixy jsou v uložení tříděny podle své délky. Tímto řešením bylo docíleno řízení rozložení počtu prefixů v závislosti na jejich délce.

K uspokojení doporučení přidělování bloků adresového prostoru EU (viz [14]) byla mezi výchozí délky prefixů zařazena délka /56.

Je třeba dodat, že náhodný generátor striktně negeneruje prefixy délky kratší než /32 včetně této, pokud se nejedná o náhodně vytvořené počáteční prefixy /32 připojené do výstupního souboru zadáním parametru. Toto chování může ovlivnit počet prefixů na výstupu.

### 7.3.4 Výstup

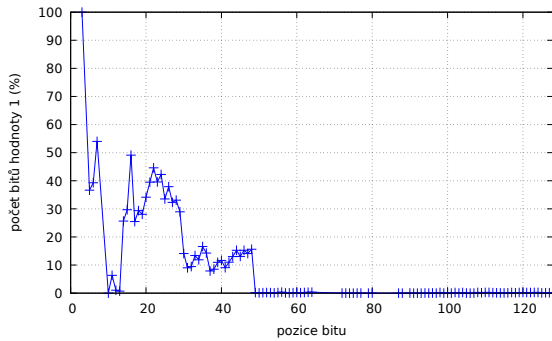
Ve výstupní fázi je předán výsledek generování na výstup dle parametru.

## 7.4 Spuštění

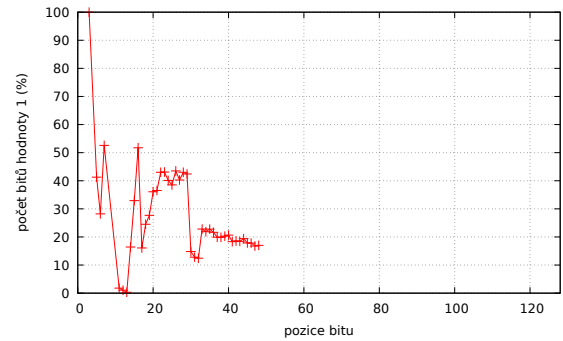
Spuštění vytvořeného generátoru probíhalo za shodných předpokladů jako u referenčního generátoru (viz 5). Z analyzované prefixové sady roku 2014 byly vybrány údaje týkající se rozložení délky prefixu včetně procentuálního zastoupení a použity jako hodnota parametru `scatter`. V tomto případě byl za vstupní soubor brán obsah prefixů zdroje Routeviews (São Paulo) [1] z roku 2011. Očekávaný výstup měl být porovnán s nejaktuálnější sadou prefixů roku 2018. Následné spuštění generátoru však začalo vykazovat chování připomínající uvážnutí, proto byl proces zastaven. Na důkaz že se o uvážnutí generátoru nejednalo byl v snížen počet požadovaných prefixů a generování spuštěno se shodnými parametry znovu. V tomto případě proběhlo generování s úspěchem. Nemělo však prakticky žádnou vypovídající hodnotu kvůli nízkému počtu vytvořených prefixů.

Následovala úprava parametrů – snížením počtu požadovaných prefixů a omezením rozptylu výsledných délek na minimální počet hodnot – a opakované spuštění. Vzhledem k upraveným parametrům nelze porovnávat hodnoty s cílovým vzorkem roku 2018, malý počet vygenerovaných prefixů (cca 8000) se od cílového v tomto případě liší zhruba o polovinu počtu. Takové porovnání by bylo zavádějící. Proto se vzorek porovnává v následující části pouze proti vzorku z roku 2014. I přesto lze pozorovat značné zkreslení výsledku vytvořeného malým počtem prefixů.

Celý experiment (`test_v2_future_2`) je možné detailně prostudovat z příložených souborů **A**, jelikož bylo jeho spuštění implementováno jako automatická úloha.

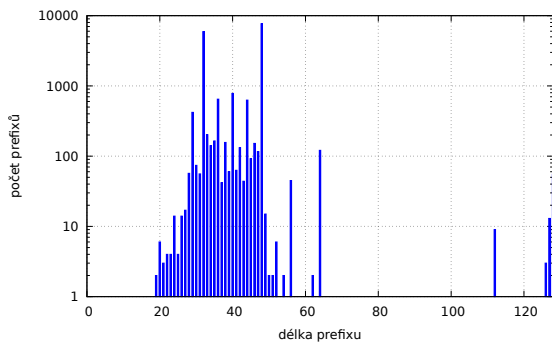


(a) Routeviews (São Paulo) 1.6.2014

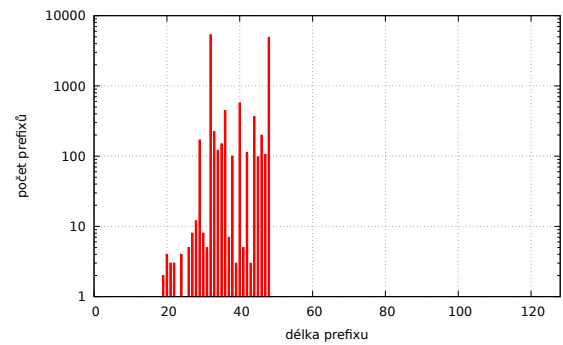


(b) Výstup generátoru

Obrázek 7.1: Porovnání rozložení bitů s hodnotou 1 v reálném vzorku z roku 2014 a výstupu implementovaného generátoru.

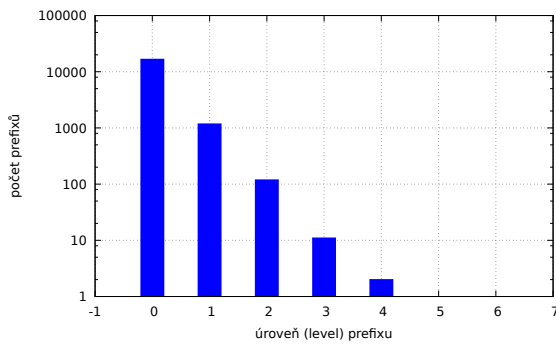


(a) Routeviews (São Paulo) 1.6.2014

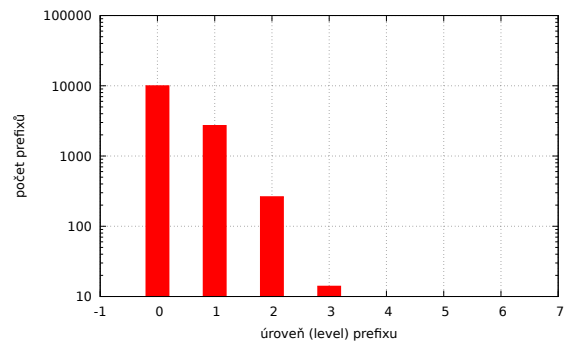


(b) Výstup generátoru

Obrázek 7.2: Porovnání rozložení délek prefixů v reálném vzorku z roku 2014 a ve výstupu implementovaného generátoru. (Povšimněte si logaritmického měřítka osy y)



(a) Routeviews (São Paulo) 1.6.2014



(b) Výstup generátoru

Obrázek 7.3: Porovnání rozložení úrovní prefixů v reálném vzorku z roku 2014 a ve výstupu implementovaného generátoru. (Povšimněte si logaritmického měřítka osy y)

# Kapitola 8

## Srovnání výsledků

V této kapitole je uvedeno srovnání dosažených výsledků generátoru 5 a vlastní implementace 7. Pro srovnání je opět možné použít pozorování závislostí v rozložení délek prefixů či rozložení hodnot bitů. Je také třeba provést diskuzi na téma efektivity obou generátorů, jejíž rozdíly se mohou zakládat na technických odlišnostech implementace.

### 8.1 Výstupy generování

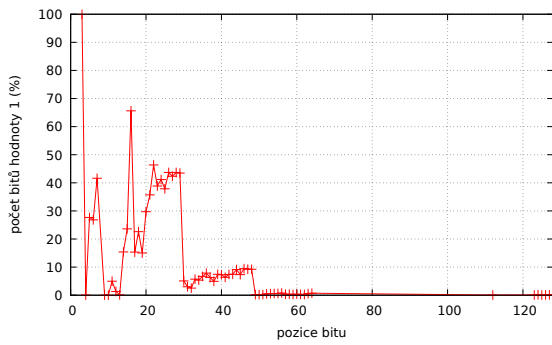
Pro demonstraci schopností obou generátorů a zároveň setrvání v mantinelech vymezených neefektivní implementací vlastního generátoru byl navržen následující experiment. Z prefixové sady Routeviews (Equinix Ashburn) [1] z roku 2010, která byla vybrána kvůli vhodnému (malému) počtu prefixů a jasných rozdílů mezi zastoupenými hodnotami, bylo ze statistik vyextrahováno rozložení prefixů a s ním spuštěno generování dle testu `test_v1_v2` z A.

#### 8.1.1 Rozložení bitů

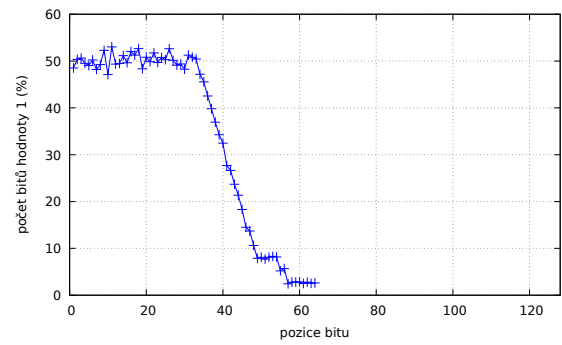
Pro srovnání na základě rozložení bitů je zásadní chování křivky v sekci a) obrázků 8.1 a 8.2 okolo délky /40 prefixu. Zatímco v prvním případě generátoru na obrázku 8.1 je vidět přibližný průběh před i za délkou /40, neplatí to pro křivku nacházející přímo nad /40. Příznivější situace nastává v případě vlastní implementace a obrázku 8.2. Pamatujeme-li na fakt, že náhodný generátor, pomocí kterého byla získána tato data není schopen generovat délky /32 a kratší pro jiné než interní účely, lze označit průběh křivky posunutý avšak velmi podobný původní křivce v sekci a). To vypovídá o schopnosti generátoru dodržet rozložení bitů.

#### 8.1.2 Rozložení délek

Na obrázcích 8.3 a 8.4 je opět důležité si všimnout v sekci a) nastaveného trendu rozložení délek prefixu. Opět musíme vzít v potaz jisté zkreslení generovaných dat v souvislosti práce náhodného generátoru s délkami /32 a menšími. V sekci a) od délky /32, již reprezentuje nevyšší špička v grafu, napravo je vidět ve srovnání prakticky shodné rozložení délek pouze na obrázku 8.4, což opět vypovídá o možnostech generátoru v závislosti na vhodném nastavení parametrů.

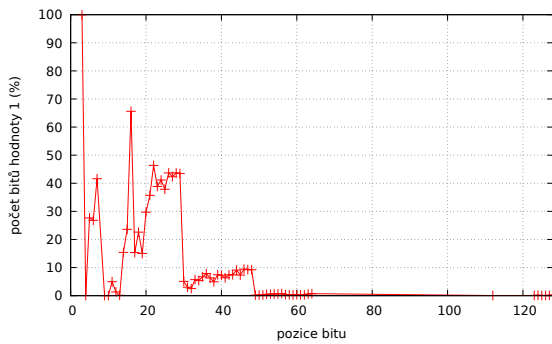


(a) Routeviews (Equinix Ashburn)

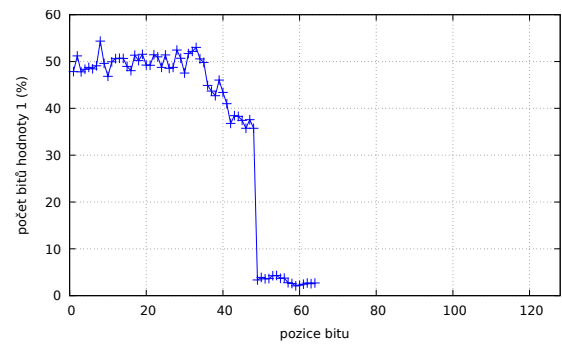


(b) Výstup generátoru 5

Obrázek 8.1: Porovnání rozložení bitů s hodnotou 1



(a) Routeviews (Equinix Ashburn)

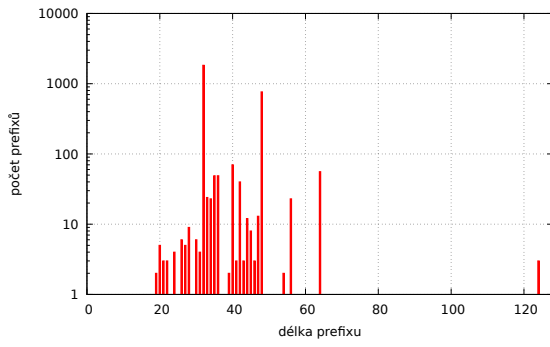


(b) Vlastní implementace 7

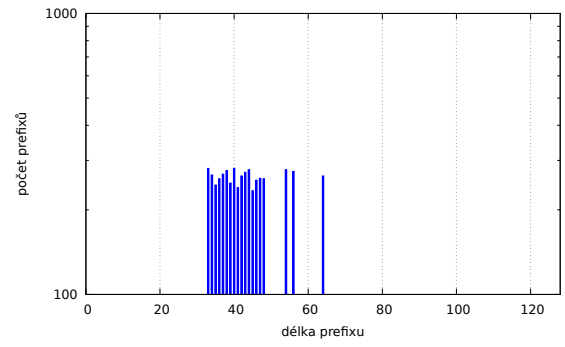
Obrázek 8.2: Porovnání rozložení bitů s hodnotou 1

### 8.1.3 Efektivita generátorů

Hlavním rozdílem mezi oběma generátory je především způsob interního ukládání prefixů. V 5 je s prefixy manipulováno jako s textovými řetězci, kdežto v 7 je každý prefix reprezentován objektem. Tento rozdíl by mohl vysvětlit chování vlastní implementace po zadání požadavku na větší množství generovaných prefixů na výstupu. A tedy zdánlivé uvážnutí (ve smyslu pojmu — deadlock) aplikace.

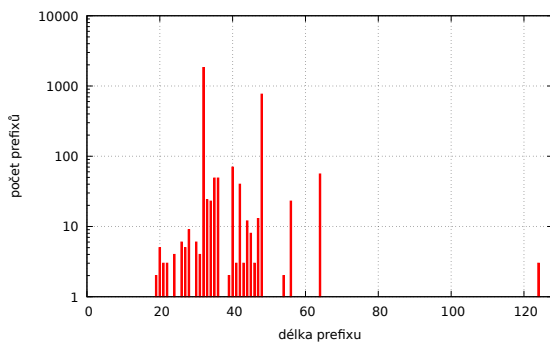


(a) Routeviews (Equinix Ashburn)

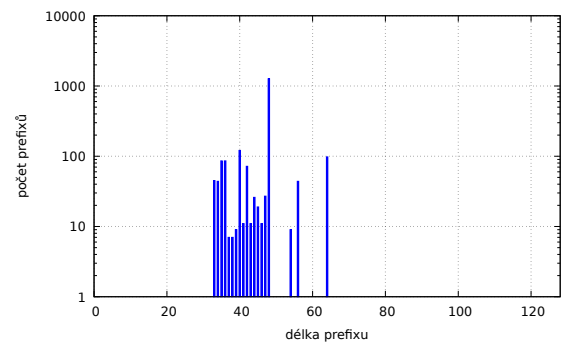


(b) Výstup generátoru 5

Obrázek 8.3: Porovnání rozložení délek prefixu



(a) Routeviews (Equinix Ashburn)



(b) Vlastní implementace 7

Obrázek 8.4: Porovnání rozložení délek prefixu

## Kapitola 9

# Závěr

Výstupem této práce měl být vlastní návrh a vytvoření fungujícího generátoru IPv6 prefixů, jenž by mohl být dále využíván k produkovaní testovacích dat nutných k vývoji vyhledávacích algoritmů směrovacích tabulek.

Implementaci nástroje k vytváření prefixů předcházelo několik dílčích úkonů. Nejdůležitější z nich bylo zjištění aktuálního stavu, porovnání existujících přístupů k problematice a v neposlední řadě provedení analýzy na základě zjištěných údajů. Kombinace znalostí získaných rešerší aktualit posloužila k vytvoření generátoru 5, jenž vycházel návrhem z článku [17].

Dalším studiem informací souvisejících s pravidly přidělování IPv6 adres a analýzou reálně dostupných prefixových sad bylo možné dospět k závěrům pro vytvoření onoho vlastního generátoru IPv6 prefixů (viz 7).

Jak shrnuje kapitola 7 samotný návrh má potenciál doručit uživateli požadovaný kvalitní výsledek, nicméně se ukázalo, že provedená implementace nepracuje efektivně s paměťovým modelem. To se projevuje hlavně v situacích, kdy je požadováno vytvoření velkého množství prefixů.

Při srovnávání výsledků metodou pozorování závislostí v rozložení bitů a délek prefixů byly v kapitole 8 nalezeny důkazy o tom, že i přes jistá omezení výsledná implementace není scestná a pokud by se v budoucnu uvažovalo o využití této aplikace, bylo by třeba zrevidovat a upravit práci s pamětí. Případně v aplikaci rozšířit možnosti náhodné části generování prefixů, která je aktuálně limitována pouze pro práci v určitém intervalu délek prefixů.

# Literatura

- [1] *University of Oregon Route Views Archive Project*. Routeviews - University of Oregon Route Views Project, Jun 2004, [Online; navštíveno 13.05.2018].  
URL <http://routeviews.org>
- [2] *Internet Assigned Numbers Authority (IANA) Policy For Allocation of IPv6 Blocks to Regional Internet Registries*. Internet Corporation for Assigned Names and Numbers (ICANN), Sep 2006, [Online; navštíveno 13.05.2018].  
URL <https://www.icann.org/resources/pages/allocation-ipv6-rirs-2012-02-25-en>
- [3] *IPv6 adoption*. Google, Jan 2009, [Online; navštíveno 13.05.2018].  
URL <https://www.google.com/intl/en/ipv6/statistics.html#tab=ipv6-adoption>
- [4] *RIS Raw Data*. RIPE NCC, Feb 2011, [Online; navštíveno 13.05.2018].  
URL <https://www.ripe.net/analyse/internet-measurements/routing-information-service-ris/ris-raw-data>
- [5] *IPv6 Address Allocation and Assignment Policy*. Réseaux IP Européens Network Coordination Centre (RIPE NCC), Mar 2018, [Online; navštíveno 13.05.2018].  
URL <https://www.ripe.net/publications/docs/ripe-699>
- [6] *Regional Internet registry*. Wikipedia, May 2018, [Online; navštíveno 13.05.2018].  
URL [https://en.wikipedia.org/wiki/Regional\\_Internet\\_registry](https://en.wikipedia.org/wiki/Regional_Internet_registry)
- [7] Bates, T.; Chandra, R.; Katz, D.; aj.: *Multiprotocol Extensions for BGP-4*. The Internet Engineering Task Force (IETF), Jan 2007, [Online; navštíveno 13.05.2018].  
URL <https://tools.ietf.org/html/rfc4760>
- [8] Coltun, R.; Ferguson, D.; Moy, J.; aj.: *OSPF for IPv6*. The Internet Engineering Task Force (IETF), Jul 2008, [Online; navštíveno 13.05.2018].  
URL <https://tools.ietf.org/html/rfc5340>
- [9] Durand, A.; Huitema, C.: *The Host-Density Ratio for Address Assignment Efficiency: An update on the H ratio*. The Internet Engineering Task Force (IETF), Nov 2001, [Online; navštíveno 13.05.2018].  
URL <https://tools.ietf.org/html/rfc3194>
- [10] Hinden, R. M.; Deering, S. E.: *IP Version 6 Addressing Architecture*. The Internet Engineering Task Force (IETF), Feb 2006, [Online; navštíveno 13.05.2018].  
URL <https://tools.ietf.org/html/rfc4291>

- [11] Huston, G.: *IPv4 Address Report*. May 2018, [Online; navštíveno 13.05.2018].  
URL <https://ipv4.potaroo.net/>
- [12] Lipner, I.; Smith, L.: *Free Pool of IPv4 Address Space Depleted*. Number Resource Organization, Feb 2011, [Online; navštíveno 13.05.2018].  
URL <https://www.nro.net/ipv4-free-pool-depleted/>
- [13] Malkin, G. S.; Minnear, R. E.: *RIPng for IPv6*. The Internet Engineering Task Force (IETF), Jan 1997, [Online; navštíveno 13.05.2018].  
URL <https://tools.ietf.org/html/rfc2080>
- [14] Narten, T.; Huston, G.; Roberts, R. G.: *IPv6 Address Assignment to End Sites*. The Internet Engineering Task Force (IETF), Mar 2011, [Online; navštíveno 13.05.2018].  
URL <https://tools.ietf.org/html/rfc6177>
- [15] Rekhter, Y.; Li, T.; Hares, S.: *A Border Gateway Protocol 4 (BGP-4)*. The Internet Engineering Task Force (IETF), Jan 2006, [Online; navštíveno 13.05.2018].  
URL <https://tools.ietf.org/html/rfc4271>
- [16] Wang, M.; Deering, S.; Hain, T.; aj.: *Non-random Generator for IPv6 Tables*. IEEE, Aug 2005, ISBN 0-7803-8686-8.
- [17] Zheng, K.; Liu, B.: *V6Gene: a scalable IPv6 prefix generator for route lookup algorithm benchmark*. IEEE, Apr 2006, ISBN 0-7695-2466-4.

# Příloha A

## Obsah CD

**genie-v1** Adresář obsahuje implementaci generátoru IPv6 prefixů podle [17].

**genie-v2** Adresář obsahuje vlastní implementaci generátoru IPv6 prefixů.

**scripts** Adresář obsahuje pomocné skripty, které jsou schopné běhu v této adresářové struktuře.

**seeds** Adresář obsahuje použité IPv6 prefixové sady.

**tests** Adresář obsahuje data provedených experimentů, které jsou prezentovány v BP.

**text** Adresář obsahuje zdrojový text BP vložený do fakultní šablony.

**xobrus00-bp.pdf** Vysázený text BP ve formátu PDF.