# UNIVERZITA PARDUBICE

Fakulta elektrotechniky a informatiky


Návrh sítě malého až středního podniku s implementací IPv6


Jiří Korejtko



Bakalářská práce

2011

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: **2010/2011**

# ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

| | |
|---|---|
| Jméno a příjmení: | **Jiří KOREJTKO** |
| Osobní číslo: | **I08084** |
| Studijní program: | **B2646 Informační technologie** |
| Studijní obor: | **Informační technologie** |
| Název tématu: | **Návrh sítě malého až středního podniku s implementací IPv6** |
| Zadávající katedra: | **Katedra informačních technologií** |

Zásady pro vypracování:

Navrhněte implementaci IPv6 v lokální síti pro malý až střední podnik s provozem běžných základních služeb na protokolu IPv6 s konektivitou do Internetu.
Běžné služby představují - provoz webového, mailového, souborového a FTP serveru.
Testování proběhne na produktech firmy Cisco.

Rozsah grafických prací:

Rozsah pracovní zprávy:

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

Seznam odborné literatury:

***SPORTACK, Mark A. Směrování v sítích IP. Praha : Computer Press, 2004. 368 s. ISBN 80-251-0127-4.**

***LAMMLE, Todd. CCNA : Výukový průvodce přípravou na zkoušku 640-802. Praha : Computer Press, 2010. 928 s. ISBN 978-80-251-2359-1.**

***PETERKA, Jiří. Archiv článků a přednášek Jiřího Peterky [online]. 2001. Dostupné z WWW: http://www.earchiv.cz/.**

Vedoucí bakalářské práce: **Mgr. Josef Horálek**
Katedra softwarových technologií

Datum zadání bakalářské práce: **17. prosince 2010**

Termín odevzdání bakalářské práce: **13. května 2011**

L.S.

prof. Ing. Simeon Karamazov, Dr.
děkan

Ing. Lukáš Čegan, Ph.D.
vedoucí katedry

V Pardubicích dne 31. března 2011

**Prohlášení autora**

Prohlašuji, že jsem tuto práci 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 Pardubicích dne 12. 8. 2011 Jiří Korejtko

**Poděkování**

Rád bych tímto poděkoval vedoucímu mé bakalářské práce, Mgr. Josefu Janu Horálkovi za ochotu, cenné připomínky a rady při tvorbě tohoto díla.

**Anotace**

Cílem této práce je popsat vlastnosti a strukturu Internetového Protokolu verze 6. Probírá se zde formát datagramů IPv6, adresace, principy objevování sousedů a automatické konfigurace adres. Tato práce se dále zabývá základními praktikami směrování v IPv6 síti a metodami přechodu mezi IPv4 a IPv6. V závěru je uvedena podpora v současných zařízeních a základní nastavení Cisco směrovačů a PC s MS Windows. V praktické části je model IPv6 sítě s konektivitou do IPv4 v simulačním programu Cisco Packet Tracer.

**Klíčová slova**

IPv6, Internet Protokol, počítačová síť, směrovač, Cisco

**Title**

A network design for a small or mid-sized company and implementation of the IPv6

**Annotation**

The aim of the thesis is to describe the features and structure of the Internet Protocol version 6. It deals with the format of IPv6 datagrams, addressing, principles of neighbour discovery and automatic address configuration. Furthermore, the paper is focused on the basic methods of routing in IPv6 network and the ways of IPv4-to-IPv6 transition. Finally, it discusses the support in contemporary devices and the basic configuration of Cisco routers and computers with MS Windows. The practical part of the thesis contains a model of IPv6 network which is connected to IPv4 network in the Cisco Packet Tracer simulation program.

**Keywords**

IPv6, Internet Protocol, computer network, router, Cisco

# Obsah

## Seznam zkratek

| | |
|---|---|
| ACL | Access Control List |
| ARP | Address Resolution Protocol |
| BGP | Border Gateway Protocol |
| BOOTP | Bootstrap Protocol |
| CCNA | Cisco Networking Academy |
| CIDR | Classless Inter-Domain Routing |
| CRC | Cyclic Redundancy Check |
| DDoS | Distributed Denial of Service |
| DHCP | Dynamic Host Configuration Protocol |
| DNS | Domain Name Service |
| DoS | Denial of Service |
| DUID | DHCP Unique Identifier |
| EGP | External Gateway Protocol |
| EUI | Extended Unique Identifier |
| GNS | Graphical Network Simulator (www.gns3.net) |
| IA | Identity Association |
| IAID | Identity Association Identifier |
| IANA | the Internet Assigned Numbers Authority |
| IETF | Internet Engineering Task Force |
| IGP | Internal Gateway Protocol |
| IOS | Internetwork Operating System |
| IPsec | IP security |
| IPv4 | Internet Protocol version 4 |
| IPv6 | Internet Protocol version 6 |
| IS-IS | Intermediate System to Intermediate System |
| ISP | Internet Service Provider |
| LSA | Link State Advertisement |
| MAC | Media Access Control |
| NAT | Network Address Translation |
| NAT-PT | Network Address Translation + Protocol Translation |
| NATPT-PT | Network Address Port Translation + Protocol Translation |
| OSPF | Open Shortest Path First |
| PC | Personal Computer |
| RARP | Reverse Address Resolution Protocol |
| RIPng | Routing Information Protocol new generation |
| RFC | Request For Comments |

## Seznam obrázků



## Seznam tabulek

# 1 Úvod

Internetový protokol je základním komunikačním jazykem celého Internetu. Jeho stávající verze (IPv4), jejíž počátky sahají do osmdesátých let, pomalu dosluhuje, a to zejména proto, že již v některých regionech došly IPv4 adresy. Už na začátku devadesátých let bylo podle některých vědeckých studií zřejmé, že IPv4 adresy brzy dojdou, a proto IETF začalo pracovat na následníkovi, protokolu IPv6. Jelikož bylo na řešení relativně dost času, rozhodlo se IETF pro zásadnější koncepční změnu, která by přinesla kromě většího adresního prostoru i nové možnosti.

Podle RFC 1550 byly na nový protokol vznešeny tyto hlavní požadavky:

- dostatečná velikost adresního prostoru na dostatečně dlouhou dobu,
- přechod z IPv4 na IPv6,
- zvýšená bezpečnost,
- co nejvíce automatické konfigurace,
- mobilita (přenosné počítače, mobilní telefony, …),
- toky a zajištění kvality jejich přenosu,
- využití na stejných komunikačních mediích jako IPv4,
- a další.

V této práci se budu zabývat obecným představením protokolu IPv6 a jeho aktuální situace. Rozeberu zde formáty paketů, významy jejich jednotlivých částí, podobnost s předchůdcem IPv4. Dále popíši protokoly úzce související s funkčností IPv6 v síti jako ICMPv6. Zaměřím se také na možnosti směrování IPv6 paketů po síti a na směrovací protokoly podporující IPv6. Také se budu věnovat možnostem propojení stávajících IPv4 sítí s novými IPv6 sítěmi a ukázkám konfigurací na Cisco zařízeních.

Jako praktickou část této práce vytvořím návrh ukázkové sítě malého až středně velkého podniku s využitím tohoto protokolu.

# 2 Formát IPv6

## 2.1 Hlavička datagramu IPv6

Základním dokumentem, který specifikuje IPv6 je RFC 2460: *Internet Protocol, Version 6 (IPv6) Specification* [1], který definuje především formát datagramu IPv6 protokolu. Ostatním principům a datovým formátům jsou věnovány další RFC dokumenty. Datagram má obvyklý tvar, nejprve hlavičky, adresy a dále nesená data. Návrh byl zaměřen zejména na jednoduchost a při tvorbě byla snaha o zajištění konstantní délky hlaviček z důvodu rychlejšího zpracování na jednotlivých síťových prvcích.

V porovnání s IPv4 hlavičkami došlo v IPv6 k dosti razantním změnám. Dříve byla délka hlaviček proměnlivá, což umožnovalo připojovat další nepovinné volby. Také obsahovaly kontrolní součet, který se musel přepočítávat na každém směrovači, jímž datagram prošel. [2], [3], [4]

Základní tvar hlavičky vidíte na obrázku 1.


**Obrázek 1 - Základní tvar IPv6 hlavičky**

*Převzato z [1 str. 4], úpravy autor*

IPv6 hlavička byla značně zredukována. Její velikost je 40 bajtů, což je sice dvojnásobek IPv4 hlavičky, ale vzhledem ke čtyřnásobnému prodloužení IP adresy odesilatele a příjemce je znát radikální zmenšení. Z této velikosti zabírají 32 bajtů adresy. Na obrázku 2 je porovnání IPv4 a IPv6 hlavičky. Položky objevující se v obou protokolech jsou vyznačeny šedou barvou a pomocí čísel u každé takovéto položky je naznačeno, které spolu v jednotlivých protokolech souvisí. Význam jednotlivých částí popíšu dále. [2], [3]

### Verze (Version)

Tato položka je obvyklým zahájením datagramů většiny protokolů. Má velikost čtyři bity a v IPv6 datagramu má hodnotu 6.

### Třída provozu (Traffic class)

Třída provozu je osmibitová hodnota odpovídající položce Type of service v hlavičce datagramu IPv4. V kombinaci se zdrojovou a cílovou adresou od sebe můžeme rozlišit různé druhy provozu a přidělovat jim priority. Díky tomu lze s jednotlivými pakety na síťových prvcích zacházet různě (např. některé upřednostňovat). Cílem je, aby protokol IP mohl poskytovat služby se zaručenou kvalitou. V praxi ale toto ještě není možné. V základní definici IPv6 není tato položka nijak podrobně rozebrána, pouze je zde uvedeno, aby implicitní hodnotou byla nula.

### Značka toku (Flow label)

V dalších dvaceti bitech se nachází značka toku. Jako tok by měly být označovány datagramy, které mají stejnou cestu, tudíž stejný zdroj, cíl a stejné nastavení hop-by-hop a routovací hlavičky. To by mělo značně zrychlit směrování, protože router bude řešit směrování pouze u prvního paketu v toku, nastavení si zapamatuje a použije ho na všechny další pakety v tomto toku. Tato koncepce je v IPv6 úplnou novinkou a ještě není konkrétně specifikována a nachází se ve fázi experimentu. Existuje mnoho návrhů, které však nebyly přijaty.

### Délka dat (Payload length)

Délka dat je šestnácti bitová nezáporná hodnota udávající velikost dat za základní IPv6 hlavičkou. Zahrnuje tedy i rozšiřující hlavičky. Jelikož je tato položka dvoubajtová, maximální délka nesených dat je 64 kB. Kdybychom potřebovali přenést větší objem dat v jednom datagramu, můžeme využít možností rozšiřující Jumbo hlavičky.

### Další hlavička (Next header)

Jelikož základní IPv6 hlavička byla značně zredukována, a to zejména o volitelné parametry, IPv6 nabízí řetězení volitelných hlaviček. Tato položka určuje typ následující hlavičky. Podrobněji se rozšiřujícím hlavičkám budu věnovat v části 2.2 na straně 14.

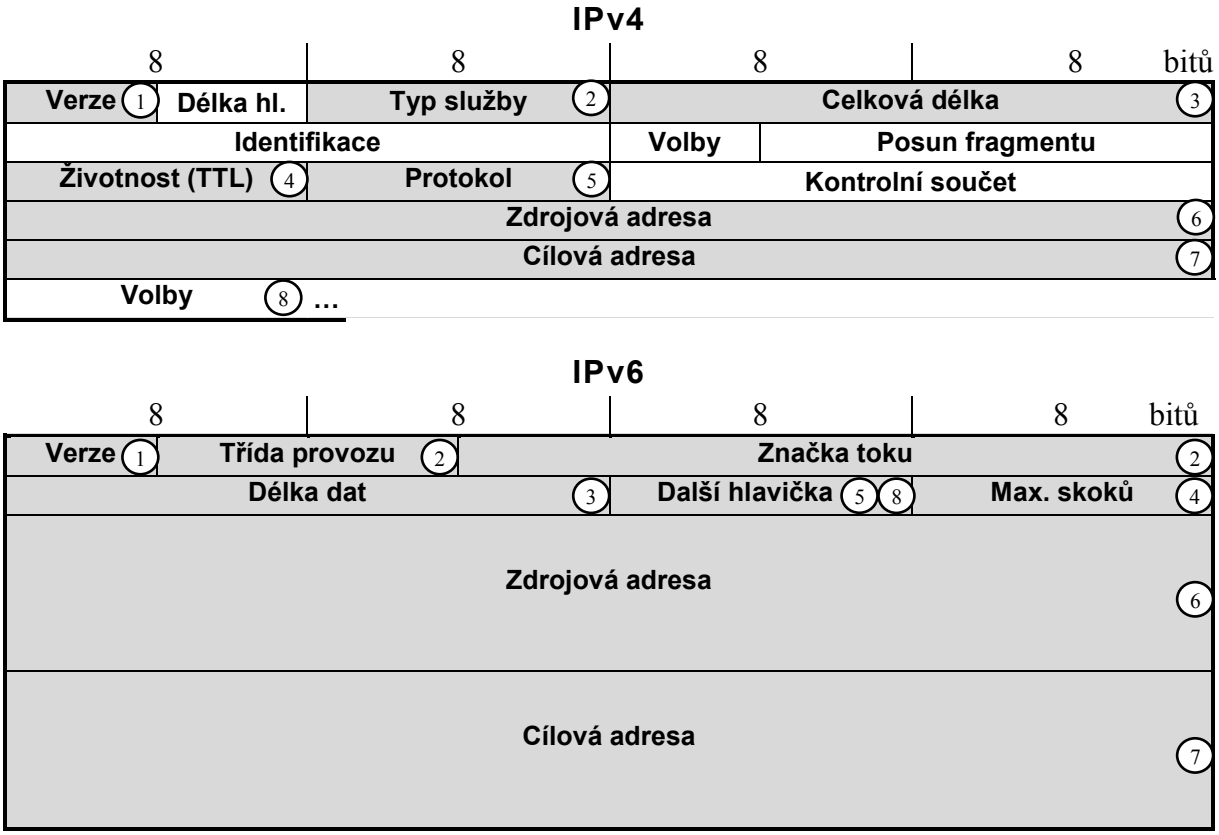
### Maximální počet skoků (Hop limit)

Tato položka nahrazuje hodnotu *Time to live* z hlavičky IPv4. Každý průchod datagramu směrovačem je považován za jeden skok. Odesilatel datagramu vždy určí, kolik skoků může datagram absolvovat. Jelikož je tato položka jednobajtová, její maximální hodnota je 255. Každý směrovač po cestě sníží tuto hodnotu o jedničku. Pokud se bude rovnat nule, paket bude zahozen a odesilateli se pošle ICMP zpráva o dosažení maximálního počtu skoků. Toto slouží jako opatření proti zacyklení paketu v síti a zbytečnému zatěžování sítě.

### Adresy (Source address a Destination address)

Dvě 128bitové položky obsahující zdrojovou a cílovou IPv6 adresu. Zdrojová adresa je vždy individuální adresa vysílajícího uzlu. Cílová adresa může být individuální

nebo skupinová adresa plánovaného příjemce. V případě použití rozšiřující směrovací hlavičky nemusí být tato adresa adresou koncového uzlu. Vzhledem ke své délce tyto dvě položky zabírají 80 % hlavičky datagramu. Podrobněji se adresaci budu věnovat v kapitole 3 na straně 21.




**Obrázek 2 - Porovnání IPv4 a IPv6 hlaviček**

*Převzato z [2 str. 35], úpravy autor*

## 2.2 Rozšiřující hlavičky

Namísto nalepování nejrůznějších voleb na konec základní hlavičky se vývojáři rozhodli využít řetězení rozšiřujících hlaviček vzájemně propojených přes položku *Další hlavička*. Každá hlavička je samostatný blok začínající položkou *Další hlavička* a *Délka hlavičky* dále pak následují podle typu jednotlivé volby. V tabulce 1 se nachází přehled nejčastějších hodnot, které se mohou objevit v položce *Další hlavička*. Jejich kompletní seznam najdete na webu organizace IANA[1]. Největší výhodou této koncepce je pružnost, jednoduchost a úspornost. Součástí datagramu jsou jen ty informace, které jsou opravdu potřeba. S tímto způsobem připojování hlaviček nastává problém, že se prodlouží čas potřebný pro nalezení informace ve větším množství hlaviček. Aby se tento čas minimalizoval, IPv6 definuje pořadí, ve kterém se hlavičky mohou připojovat. V zásadě jde o to, aby se hlavičky potřebné pro směrování připojovaly jako první a hlavičky určené pro adresáta byly na konci řetězce. Například pro průchozí směrovač jsou důležité

[1] http://www.iana.org/assignments/protocol-numbers/

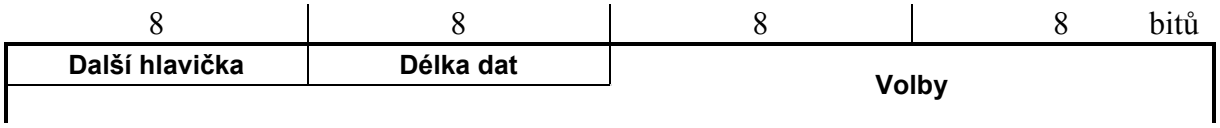
informace v hlavičce *Volby pro všechny*, která se smí objevit jen bezprostředně za základní hlavičkou. Pokud tedy směrovač narazí v základní hlavičce v poli *Další hlavička* na jiné číslo než nulu, ví, že už v datagramu nenajde žádné pro něj užitečné informace a může s analýzou skončit. Speciální význam má hodnota 59 (No next header), která značí, že se jedná o poslední hlavičku a za ní již není nic. Pokud se v datagramu podle jeho délky mají nacházet ještě nějaká data, musí být ignorována. Je-li datagram předáván dále, musí ho směrovač přeposlat beze změny i s přebytečnými daty. [2], [3], [4]

**Tabulka 1 - Rozšiřující hlavičky a protokoly**

| Hodnota | Rozšiřující hlavička |
|---|---|
| 0 | Volby pro všechny (Hop-by-hop option) |
| 43 | Směrování (Routing) |
| 44 | Fragmentace (Fragment) |
| 50 | Šifrování obsahu (ESP) |
| 51 | Autentizace (AH) |
| 59 | Poslední hlavička (No next header) |
| 60 | Volby pro cíl (Destination option) |
| 135 | Mobilita (Mobility header) |
| | **Protokoly** |
| 6 | TCP |
| 8 | EGP |
| 9 | IGP |
| 17 | UDP |
| 46 | RSVP |
| 47 | GRE |
| 58 | ICMPv6 |

## 2.3 Rozšířené volby

Hlavičky, které obsahují volby, jsou v IPv6 definovány dvě: *Volby pro všechny* (hop-by-hop options) a *Volby pro cíl* (destination options). Obě dvě hlavičky mají stejnou podobu, která je znázorněna na obrázku číslo 3. Položka *Další hlavička* má totožný význam jako v základní hlavičce. *Délka dat* určuje velikost všech voleb v hlavičce v násobcích bajtů. Do délky se nezapočítává prvních osm bitů. To znamená, že když je v položce *Délka dat* uvedena hodnota 1, celková délka rozšiřující hlavičky je 16 bitů. Poslední položka jsou samotné *Volby*. Tato položka je omezena pouze maximální hodnotou délky dat. V RFC 2460 jsou definovány jen dvě volby: Pad1 a PadN. [1], [2], [3]




**Obrázek 3 - Formát rozšiřujících hlaviček *Volby pro všechny* a *Volby pro cíl***

*Převzato z (1 str. 11), úpravy autor*

V tabulkách 2 a 3 můžete vidět přehled nejčastějších voleb. Kompletní seznam voleb lze najít na internetových stránkách organizace IANA[2].

Pad1 a PadN jsou speciální volby využívané k zarovnání ostatních voleb do čtyřbajtových bloků. Nenesou žádnou informaci.

**Tabulka 2 - Přehled nejčastějších voleb pro všechny**

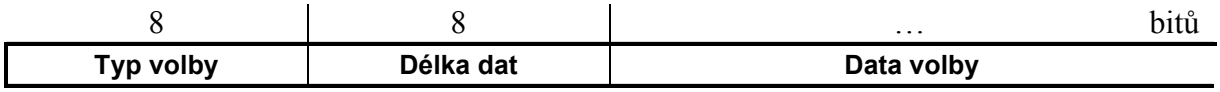
| Hodnota | Význam |
|---|---|
| 0 | Pad1 |
| 1 | PadN |
| 5 | Upozornění směrovače |
| 194 | Jumbo obsah |

**Tabulka 3 - Přehled nejčastějších voleb pro cíl**

| Hodnota | Význam |
|---|---|
| 0 | Pad1 |
| 1 | PadN |
| 201 | Domácí adresa |

### 2.3.1 Formát voleb

Všechny volby musí dodržovat formát definovaný IPv6, který je znázorněn na obrázku 4. První bajt určuje, o jakou volbu se jedná. Další položkou je *Délka dat*, která uvádí délku voleb bez prvních dvou bajtů. Nakonec následují data, která musí přesněji specifikovat dokument, který zavede danou volbu.




**Obrázek 4 - Formát voleb pro rozšiřující hlavičky**

*Převzato z [1 str. 9], úpravy autor*

V definici IPv6 je přesně uveden význam prvních tří bitů typu volby. První dva určují, jak má zařízení, které dané volbě nerozumí, s paketem naložit. Třetí bit určuje, zda je možné volbu během cesty k cíli měnit, či nikoliv. Pokud je mezi rozšiřujícími hlavičkami autentizační hlavička, všechny volby, které mají nastaveno, že je možné je měnit, musí být považovány za nulové. [1]

Možnosti nastavení dvou nejvyšších bitů:

00 – Přeskočit tuto volbu a pokračovat ve zpracování hlavičky.

01 – Zahodit paket.

10 – Zahodit paket a odeslat zdroji ICMP zprávu o neznámém typu hlavičky bez ohledu na to, zda je cílová adresa individuální nebo skupinová.

[2] http://www.iana.org/assignments/ipv6-parameters/

11 – Zahodit paket, a pokud není cílová adresa skupinová, odeslat zdroji ICMP zprávu o neznámém typu hlavičky.

Možnosti nastavení třetího nejvyššího bitu:

0 – Data není možno po cestě upravovat.

1 – Data mohou být po cestě změněna.

### 2.3.2 Pad1

Pad1 je jednobajtová volba speciálního formátu. Neobsahuje položky Délka dat a žádné hodnoty voleb. Její hodnota je nulová. Pokud potřebujeme vyplnit více místa než jeden bajt, je doporučeno použít položku PadN místo několika položek Pad1. [1], [3]

### 2.3.3 PadN

Volba PadN je využívána je-li potřeba zarovnat dva nebo více bajtů. Typ této volby je 1. Druhý bajt obsahuje délku volby v násobcích bajtů. Do délky se ale nezapočítávají první dva bajty, takže pokud budeme chtít vynechat například 5 bajtů, *Délka dat* bude obsahovat hodnotu 3 a za ní budou následovat tři bajty nulových hodnot. [1], [3]

### 2.3.4 Upozornění směrovače

Jednou z nejdůležitějších voleb je upozornění směrovače. Tato volba je definována v RFC 2711: *IPv6 Router Alert Option* [5]. Cílem této volby je umožnit směrovačům jednoduše a rychle rozpoznat, jestli paket nese data, která by mohla směrovač zajímat. Formát volby upozornění směrovače lze vidět na obrázku 5.

| 8 | 8 | 8 | 8 bitů |
|---|---|---|---|
| **Typ volby = 5** | **Délka dat = 2** | **Hodnota (protokol)** | |

**Obrázek 5 - Volba upozornění směrovače**

*Převzato z [5 str. 2], úpravy autor*

Upozornění směrovače se uplatní například u rezervačního protokolu RSVP nebo při provozování skupinových služeb v síti. V tabulce 4 se nacházejí hodnoty, kterých může upozornění směrovače nabývat. [5]

**Tabulka 4 - Přehled hodnot pro volbu Upozornění směrovače**

| Hodnota | Význam |
|---|---|
| 0 | Datagram obsahuje MLD zprávu |
| 1 | Datagram obsahuje RSVP zprávu |
| 2 | Datagram obsahuje zprávu aktivní sítě |
| 3-65535 | Hodnoty rezervované IANA k budoucímu použití |

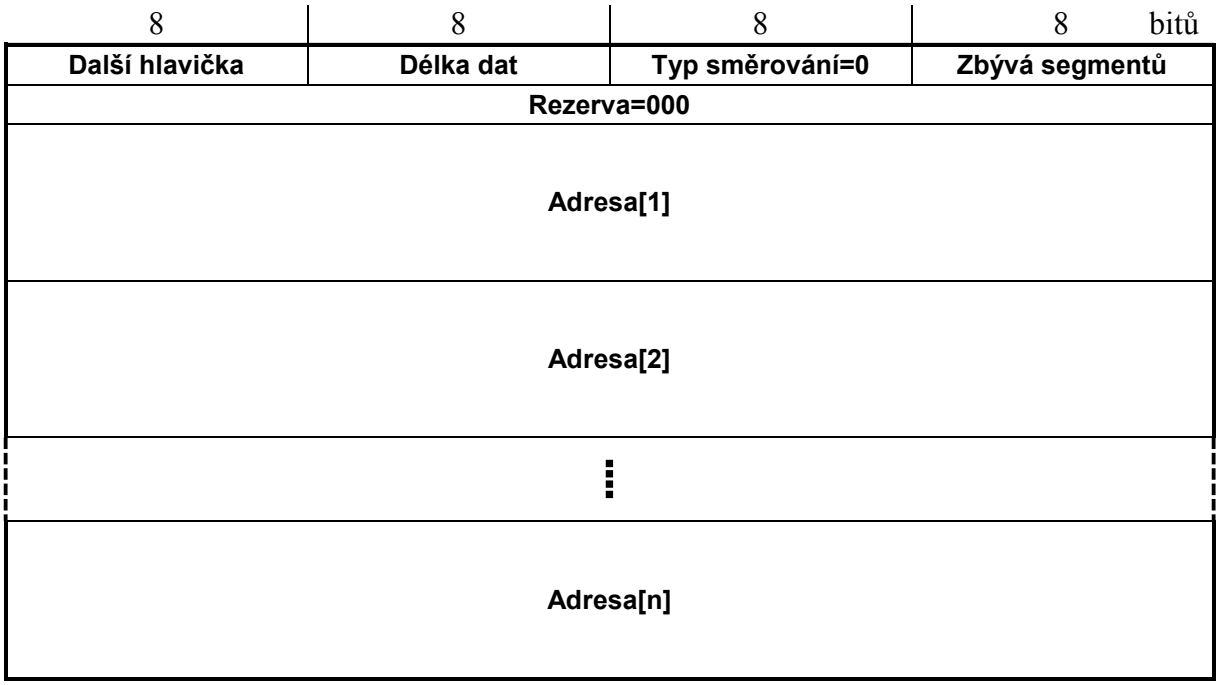
## 2.4 Směrovací hlavička

Směrovací hlavičkou jde upravit standardní způsob směrování, které se řídí pouze cílovou adresou uzlu. Je možné určit několik adres, přes které musí datagram projít předtím, než dorazí k adresátovi. V IPv6 máme možnost využít několik různých typů směrovacích hlaviček, které se rozlišují hodnotou položky *Typ směrování*.

V současné době jsou definovány a přesně popsány pouze dva typy směrování. Samotné IPv6 definuje směrování typu 0 a v RFC6275: *Mobility Support in IPv6* je definován typ 2, zjednodušené směrování využívané pro mobilitu. Další dva definované typy (253 a 254) jsou určeny pro testovací a experimentální účely. [2]

### 2.4.1 Směrování typu 0

Směrovací hlavička typu 0 nese dvě důležité informace: seznam adres, které má datagram navštívit a počet uzlů, které zbývá ještě projít. Tyto dvě informace spolu tvoří na začátku plán cesty sítí a nakonec záznam adres, přes které datagram prošel. Chce-li odesilatel, aby jím odeslaný datagram prošel určitými uzly, využije tuto směrovací hlavičku. Jako adresáta datagramu uvede první uzel, kterým chce, aby datagram prošel. Přidá směrovací hlavičku typu 0 a do seznamu adres uvede všechny zbývající uzly. Adresáta datagramu uvede jako posledního do seznamu adres. Položku *Zbývá segmentů* vyplní počtem adres v seznamu a datagram odešle obvyklým způsobem. Při použití tohoto typu směrování nesmí být použita žádná skupinová adresa [2], [3]. Formát směrovací hlavičky je znázorněn na obrázku 6.

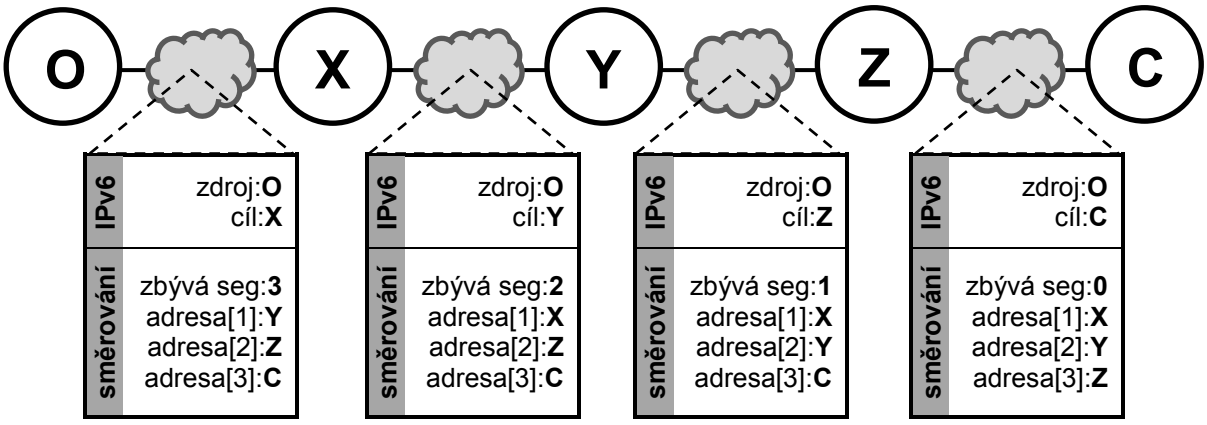



**Obrázek 6 - Formát hlavičky směrování typu 0**

*Převzato z [1 str. 14], úpravy autor*

Když je datagram doručen na cílovou adresu uvedenou v základní IPv6 hlavičce (v našem případě první průchozí uzel), router analyzuje směrovací hlavičku a zjistí, že je jen průchozí uzel. Dále směrovač vezme ze seznamu n-tou[3] adresu od konce a vymění ji za stávající cílovou adresu. Nakonec sníží hodnotu položky *Zbývá segmentů* o jedničku a odešle datagram novému cíli. Tento proces probíhá na každém uzlu uvedeném v seznamu. Položka *Zbývá segmentů* rozděluje seznam adres na ty, kterými datagram již prošel, a na adresy, které ho ještě čekají. Nachází-li se v této položce nula, datagram dorazil do cíle [2]. Na obrázku číslo 7 se můžete podívat na důležité změny ve směrovací hlavičce při cestě od odesilatele O přes uzly X, Y a Z až k adresátovi C.

Tento typ směrování byl v RFC 5095: *Deprecation of Type 0 Routing Headers in IPv6* zavrhnut. Zejména z důvodu bezpečnosti. Pomocí tohoto typu směrovacích hlaviček je možné zahltit síť datovými toky mnohem většími, než je kapacita útočníkovy linky. Podle tohoto dokumentu musí zařízení, které obdrží datagram s touto hlavičkou, hlavičku ignorovat, pokud je počet zbývajících segmentů nulový. Pokud je počet zbývajících segmentů nenulový, musí datagram zahodit a oznámit odesilateli pomocí ICMP chybný typ směrovací hlavičky. [2], [4]




**Obrázek 7 - Změny ve směrovací hlavičce během cesty**

*Převzato z [3 str. 12]*

### 2.4.2 Směrování typu 2

Mobilita IPv6 definuje v RFC 6275: *Mobility Support in IPv6* [6] směrovací hlavičku typu 2. Jde o značně zjednodušenou podobu základního typu 0. V zásadě jde o to, že místo celého seznamu adres může hlavička obsahovat pouze jednu adresu, a to vždy domácí adresu cílového uzlu. Toto omezení značně snižuje možnost zneužití tohoto typu směrovací hlavičky.

Mobilní uzel na cestách má kromě své pevné (domácí) adresy ještě adresu dočasnou, která se odvíjí od aktuální sítě a může se měnit například jen při přecházení

[3] Kde n je aktuální hodnota položky *Zbývá segmentů*

mezi bezdrátovými přípojnými body. Aby během přesunu mobilního uzlu nebyla narušena funkčnost běžících aplikací, uzel používá pro jejich síťovou komunikaci domácí adresu.

Odesilatel umístí do cílové adresy základní hlavičky IPv6 dočasnou adresu mobilního uzlu a připojí směrovací hlavičku typu 2 s domácí adresou mobilního uzlu. Datagram dorazí na aktuální dočasnou adresu mobilního uzlu, kde se obdobným postupem jako u typu 0 vymění cílová adresa za adresu ve směrovací hlavičce a vyšším vrstvám se doručí datagram tak, jakoby přišel na domácí adresu. Formát hlavičky je obdobný jako u typu 0, jen se zde nachází pouze jedna adresa. [2]

# 3 Adresace IPv6

Nejhlavnějším důvodem pro vytvoření protokolu IPv6 byl zmenšující se adresní prostor IPv4, proto IPv6 přichází, kromě mnoha vylepšených vlastností, s adresním prostorem, který obsahuje 340 bilionů bilionů bilionů unikátních adres. Délka IPv6 adresy je 128 bitů, na rozdíl od pouhých 32 bitů u IPv4. Tento rozsah je pro blízkou budoucnost téměř nekonečný. Základní pravidla adresace definuje dokument RFC 4291: *IP Version 6 Addressing Architecture* [7] a několik dalších RFC, které definují konkrétní formáty adres, například. [8], [9]

## 3.1 Zápis adres

IPv6 adresy jsou čtyřikrát delší než adresy IPv4, proto se již nezapisují v dekadické soustavě, ale v hexadecimální. Adresa je rozdělena do osmi bloků, které jsou spojeny dvojtečkou. Takto dlouhá adresa je pro člověka téměř nezapamatovatelná, proto se mnohem více začnou využívat doménová jména, nejspíš i v lokálních sítích. K jednoduššímu zapamatování a zpřehlednění adres je možno využít několika druhů zjednodušených zápisů. [2], [7], [10]

### 3.1.1 Zkrácené zápisy

Vzhledem k častému výskytu nul v adresách se nám nabízí dva způsoby zkracování. Za prvé můžeme v každém bloku vypustit libovolný počet počátečních nul, takže místo „0000“, můžeme psát pouze „0“ nebo místo „0009“ jen „9“. Jelikož se někdy vyskytuje i několik nulových bloků za sebou, je tyto bloky možné nahradit zápisem čtyřtečky „::“ (dvě dvojtečky) [7]. Například adresu

0123:0000:0000:4500:0000:0000:0000:67AB

můžeme zkrátit na

123:0:0:4500:0:0:0:67AB

nebo i vypustit nulové bloky takto

123::4500:0:0:0:67AB

anebo ještě lépe takto

123:0:0:4500::67AB

Nahrazení dvojitou dvojtečkou lze v adrese použít jen jednou, protože by zápis ztratil jednoznačnost. Stejně tak nelze vynechat nuly za číslem „45“, protože by to znamenalo „0045“ místo „4500“. Za nejkratší by se dal určitě považovat zápis nedefinované adresy

0000:0000:0000:0000:0000:0000:0000:0000,

kterou lze zkrátit na „::“.

### 3.1.2 IPv4-mapované adresy

Pro potřeby některých přechodových mechanizmů je třeba vyjádřit pomocí IPv6 staré IPv4 adresy. K tomu se využívají takzvané IPv4-mapované adresy. Jejich formát je následující: prvních 80 bitů je nulových, dalších 16 bitů obsahuje samé jedničky a do posledních 32 bitů se umístí IPv4 adresa. Například pro IP adresu 195.113.124.150 je zápis následující

::FFFF:C371:7C96

V praxi lze použít i jednodušší způsob s tečnovým zápisem IPv4 adresy a bez přepočítávání.

::FFFF:195.113.124.150

Více informací o těchto adresách naleznete v [7].

### 3.1.3 Prefixy

Prefixy v IPv6 jdou ve stopách IPv4 prefixů definovaných podle CIDR (Classless Inter-Domain Routing). Prefixy se zapisují následujícím způsobem:

IPv6_adresa/délka_prefixu

Délka prefixu určuje, kolik začátečních bitů adresy je považováno za prefix a co za adresu uzlu. Prefixem se vyjadřuje příslušnost uzlu k určité síti nebo podsíti. Všechna rozhraní v této síti by měla mít stejný prefix, jehož délka může být různá, ale zpravidla směrem ke koncovému uživateli se prefix prodlužuje. Například šedesátibitový prefix 20010DB80000CD3 lze zapsat následovně [2]:

2001:0DB8:0000:CD30:0000:0000:0000:0000/60
2001:DB8::CD30:0:0:0:0/60
2001:DB8:0:CD30::/60

Tento prefix nelze zapsat níže uvedenými zápisy:

2001:0DB8:0:CD3/60
: Toto je chybný zápis adresy. Adresa nemá správnou délku. Nulové bloky nelze absolutně vynechat.

2001:0DB8::CD30/60
: Podle tohoto zápisu se adresa rozvine na 2001:0DB8:0000:0000:0000:0000:0000:CD30.

2001:0DB8:0:CD3::/60
: Podle tohoto zápisu se adresa rozvine na 2001:0DB8:0000:0CD3:0000:0000:0000:0000. Nelze oříznout polední nulu v bloku „CD30“, ořezávat lze pouze nuly na začátku bloku.

## 3.2 Rozdělení adres

Adresní prostor IPv6 byl rozdělen na několik skupin pro různé účely, podobně jako v IPv4. Tyto skupiny se rozlišují pomocí prefixu adresy. Byly definovány také dvě adresy se specifickým významem. První je nedefinovaná adresa „::0“, která značí, že rozhraní dosud nebyla přidělena žádná adresa. Druhou je obdoba IPv4 adresy 127.0.0.1, adresa „::1“ pro rozhraní loopback, kterým počítač může komunikovat sám se sebou. [2], [3]

**Tabulka 5 - Rozdělení IPv6 adres**

| Prefix | Význam |
|---|---|
| ::/128 | Nedefinovaná adresa |
| ::1/128 | Lokální smyčka (loopback) |
| FF00::/8 | Skupinové adresy |
| FE80::/10 | Individuální lokální linkové adresy |
| FC00::/7 | Unikátní individuální lokální adresy |
| ostatní | Individuální globální adresy |

V tabulce 5 vidíte rozdělení adres do jednotlivých skupin. Kompletní seznam přidělení adres do skupin můžete najít na stránkách organizace IANA[4]. Jak je vidět největší část zabírají individuální globální adresy. To jsou celosvětově jednoznačné adresy, z nichž by mělo mít každé rozhraní jednu přidělenu. Zatím je využíván pouze prefix 2000::/3, ostatní jsou rezervovány pro pozdější využití. Předpokládá se, že některé z dalších RFC jim přidělí nějaký význam a strukturu. [2]

### 3.2.1 Globální individuální adresy

Tyto adresy jsou z celého IPv6 rozsahu ty nejdůležitější. Je to obdoba veřejných adres IPv4 rozsahu. Každé zařízení s touto adresou je jednoznačně identifikováno v celém Internetu. První část, co byla zatím přidělena (prefix 2000::/3, binárně „001“), definuje RFC 3587: *IPv6 Global Unicast Address Format* [8]. Aktuální stav přidělení globálních individuálních adres lze sledovat na stránkách organizace IANA[5]. [2], [10]

Globální individuální adresy jsou přidělovány hierarchicky organizací IANA regionálním registrátorům (RIR, pro Evropu je to RIPE NCC). Regionální registrátor poté přiděluje adresy lokálním registrátorům (LIR), což jsou nejčastěji poskytovatelé připojení k Internetu a ti přidělují části svých prefixů koncovým zákazníkům. Tento koncept členění je pro fungování Internetu velice důležitý, zmenšuje objem dat ve směrovacích tabulkách páteřních směrovačů. Struktura adresy je obdobná jako u IPv4, které mělo adresu sítě, podsítě a rozhraní v podsíti. IPv6 má obdobné časti s upravenými názvy, jen místo názvu sítě je globální prefix [3]. Tvar těchto adres ilustruje obrázek číslo 8.

[4] http://www.iana.org/assignments/ipv6-address-space/
[5] http://www.iana.org/assignments/ipv6-unicast-address-assignments/

| 3 | 45 | 16 | 64 bitů |
|---|---|---|---|
| 001 | **Globální prefix** | **Subnet ID** | **Interface ID** |

**Obrázek 8 - Tvar globální individuální adresy**

*Převzato z [8 str. 3], úpravy autor*

### Globální prefix

Tato část adresy identifikuje celou koncovou síť v Internetu. Prefix je přidělován z rozsahu lokálních registrátorů, většinou poskytovatelů internetového připojení. Často se této částí adresy říká „veřejná topologie“. [2]

### Subnet ID

Identifikátor podsítě je již záležitostí koncové sítě, slouží k identifikaci podsítí v rámci dané sítě. Pro tuto část adresy se často používá označení „místní topologie“. Její velikost je 2 bajty, což správcům dává k dispozici 65 536 podsítí a to je číslo co by mělo bohatě vystačit i pro hodně velké sítě. Díky možnosti využít tolik podsítí si můžeme dovolit ponechat délku této části adresy konstantní. [2]

Nad tímto přesným přidělováním prefixu 48 se stále vedly diskuze, jestli to není pro běžné uživatele plýtvání. Z toho postupně vzešel dokument RFC 6177: *IPv6 Address Assignment to End Sites*, který říká, že přidělování prefixu 48 není povinné a rozhodnutí záleží na lokálním registrátorovi.

### Interface ID

Identifikátor rozhraní je největší část IPv6 adresy. Jeho velikost je 64 bitů, čímž umožňuje v jedné podsíti rozlišit miliardy miliard adres[6] pro různá rozhraní. Toto se může zdát jako nehorázné plýtvání s adresami, ale RFC 4291 to zavádí kvůli maximálnímu zjednodušení automatické konfigurace. Podle tohoto dokumentu musí všechny individuální adresy používat identifikátor rozhraní ve tvaru modifikovaného EUI-64 o délce 64 bitů.

### 3.2.2 Interface ID – Modifikované EUI-64

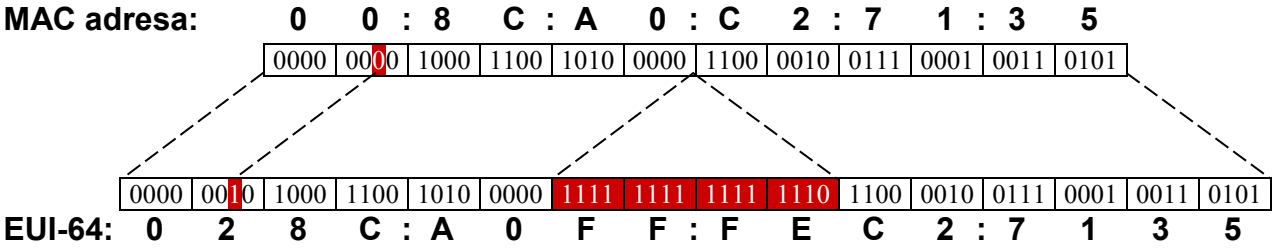
Modifikované EUI-64 je v IPv6 celosvětově jednoznačný identifikátor síťového rozhraní. Jeho velikost je 64 bitů, čímž odpovídá délce rezervovaného místa v IPv6 adrese. V EUI-64 nese předposlední bit v nejvyšším bajtu (7. bit zleva) speciální informaci, a to příznak globality. Ve standardním EUI-64 hodnota 0 znamená globálně jednoznačnou adresu a hodnota 1 adresu lokální. [2]

IPv6 používá modifikované EUI-64, kde jsou tyto hodnoty obráceny, a to z důvodu jednoduššího vytváření lokálních identifikátorů. Díky této úpravě mohou správci využívat, například pro sériové spoje, jednoduché lokální identifikátory jako

[6] Přesně $2^{64}$ = 18 446 744 073 709 551 616

1 a 2. Kdyby k této úpravě nedošlo, museli by použít místo jednoduchých identifikátorů tyto: 200:0:0:1 a 200:0:0:2. [3]

EUI-64 identifikátory se nejčastěji budou tvořit na rozhraních, které již mají přidělenou celosvětově jedinečnou MAC adresu o délce 48 bitů. V takovém případě se MAC adresa velmi snadno podle standardního algoritmu transformuje na EUI-64. Doprostřed (mezi třetí a čtvrtý bajt) MAC adresy se vsune hodnota FFFE a obrátí se příznak globality. Tedy pro MAC adresu 00:8C:A0:C2:71:35 je modifikované EUI-64 028C:A0FF:FEC2:7135. Postup vytváření identifikátorů rozhraní podle modifikovaného EUI-64 je názorně předveden na obrázku 9.




**Obrázek 9 - Vytvoření EUI-64 z MAC adresy**

*Převzato z [3 str. 18], úpravy autor*

Používaní EUI-64 je velmi jednoduché a efektivní, bohužel ale ne tak bezpečné. Jelikož je EUI-64 odvozeno od MAC adresy, která je celosvětově jednoznačná a je pevně spjata se síťovou kartou, kterou v počítači měníte jen zřídka, je možné jednoznačně identifikovat počítač v celém Internetu. U serverů to nevadí, ty mají i teď přiřazeny celosvětově jednoznačné IPv4 adresy, ale u klientských počítačů, zejména notebooků nastává problém. Tato část adresy se podle MAC adresy generuje stále stejně, ať jsme připojeni do jakékoliv sítě. Počítač zjistí prefix sítě a k němu připojí svůj identifikátor rozhraní. Pokud by se nějakému útočníkovi podařilo odposlouchávat provoz na strategických místech, mohl by jednoduše identifikovat data z vašeho počítače a sledovat s kým počítač komunikuje a v jaké síti se právě nachází. V tomto případě nepomůže ani šifrování, protože se šifrují pouze nesená data a ne hlavička datagramu. Adresy musejí zůstat čitelné pro všechny. [2], [3]

Tento problém řeší dokument RFC 4941: *Privacy Extensions for Stateless Address Autoconfiguration in IPv6* [11], který navrhuje, aby každé rozhraní mělo mimo pevného identifikátoru podle EUI-64 také jeden nebo více dočasných identifikátorů, které budou generovány náhodně a budou mít krátkou platnost (řádově od hodin do několika dnů). Tyto dočasné adresy se budou používat pouze pro odchozí komunikaci uzlu. Veškerá komunikace inicializovaná zvenčí bude směřována na adresu s pevným identifikátorem podle EUI-64, která bude zavedena i do DNS. Adresy s dočasnými identifikátory samozřejmě nebudou zavedeny do DNS, jinak by celá snaha přišla vniveč, protože by

bylo možné počítač rozpoznat podle shodného doménového jména. Díky tomuto zabezpečení nelze dlouhodobě sledovat aktivity jednotlivých počítačů.

### 3.2.3 Lokální adresy

Už v IPv4 byl zaveden koncept adres, které nejsou celosvětově jednoznačné a směly se používat jen v lokálních sítích, které nebyly připojeny k Internetu, nebo byly připojeny NATem.

V IPv6 je definováno několik druhů lokálních adres, jejich přehled je na obrázku číslo 10.

**Lokální linkové (FE80::/10)**

| 10 | 54 | 64 |
|---|---|---|
| 1 1 1 1 1 1 1 0 1 0 | 0 0 0 0 0 0 0 0 0 0 … 0 0 0 0 0 0 0 0 0 0 | **EUI-64** |

**Lokální místní (FEC0::/10) – již odmítnuté**

| 10 | 54 | 64 |
|---|---|---|
| 1 1 1 1 1 1 1 0 1 1 | **Identifikátor podsítě …** | **EUI-64** |

**Unikátní lokální (FC00::/7)**

| 7 | | 40 | 16 | 64 |
|---|---|---|---|---|
| 1 1 1 1 1 1 1 | L | **Globální identifikátor** | **Identifikátor podsítě** | **EUI-64** |

**Obrázek 10 - Typy lokálních adres**

*Převzato z [7 str. 11] a [9 str. 3], úpravy autor*

#### *3.2.3.1 Lokální linkové adresy*

Asi největší význam mají lokální linkové adresy, jejichž tvar je následující: začínají prefixem FE80::/10, dalších 54 bitů adresy je nulových a na konci je místo pro identifikátor rozhraní podle modifikovaného EUI-64. Tvorba adres je standardní, tudíž si adresu může vygenerovat každý počítač sám a pak jen pomocí nástrojů automatické konfigurace ověřit, zda je adresa v dané síti opravdu unikátní.

Například počítač s rozhraním s MAC adresou 00:8C:A0:C2:71:35 by tomuto rozhraní přidělil adresu:

FE80::28C:A0FF:FEC2:7135

Dosah těchto adres je omezen pouze na jednu linku, tudíž pakety s takovouto adresou se nedostanou přes směrovač dál, protože se za ním už vyskytují jiné linky. Oficiálně je sice prvních 64 bitů interpretováno jako adresa sítě a podsítě, ale směrovat podle nich nelze, protože je mají všechny sítě stejné.

Hlavní výhodou těchto adres je, že si je host dokáže vygenerovat úplně sám a nepotřebuje k tomu žádný server nebo směrovač. Stačí propojit počítače přes nějaký

přepínač a máte jednoduchou funkční síť. V takovéto síti ale nebude žádný DNS server, takže bude nutné zadávat nepřehledné IPv6 adresy ručně.

Těchto adres využívají i některé mechanizmy IPv6. Například DHCPv6 používá, ještě před přidělení adresy, ke komunikaci klienta se serverem lokální linkové adresy.

Další podrobnosti o tomto typu adres najdete v [2] a [10].

#### 3.2.3.2 Lokální místní adresy

Tyto adresy s prefixem FEC0::/10 měly podobný účel jako adresy lokální linkové, ale byly omezeny ne na jednu linku, ale na „místo“. Jako „místo“ se většinou považovala koncová síť jedné organizace připojená k Internetu. [2]

Jenže existuje mnoho organizací, které mají například pobočky na více místech (v různých městech nebo i státech) a nebylo jednoznačné, jestli jako „místo“ definovat celou síť organizace nebo každou pobočku zvlášť. V praxi se také objevily problémy s konfigurací směrovačů.

Výsledkem bylo zamítnutí lokálních místních adres v dokumentu RFC 3879: *Deprecating Site Local Addresses*. Tento dokument dokonce zakazuje v nových implementacích podporovat speciální zacházení s adresami s prefixem FEC0::/10. [12]

#### 3.2.3.3 Unikátní lokální adresy

Unikátní lokální adresy byly definovány v RFC 4193: *Unique Local IPv6 Unicast Addresses* [9] jako nástupce lokálních místních adres. Význam adres je obdobný jako byl u lokálních místních adres. Organizace má více poboček na různých místech a ráda by je propojila kromě globálních unikátních adres i lokálními adresami s vlastním prefixem pro celou síť. Jak je vidět na obrázku číslo 10, tyto adresy začínají prefixem FC00::/7, za ním se nachází jednobitový příznak *L*, který nese informaci o tom, zda byl prefix adresy přidělen lokálně (L=1) nebo jinak[7]. V současné době jsou všechny takovéto adresy generovány lokálně, tudíž mají nastaven příznak *L* na jedničku a jejich prefix vypadá následovně: FD00::/8. [2]

Další částí adresy je čtyřicetibitový globální identifikátor, kterým by mělo být náhodné číslo[8]. V RFC 4193 se důrazně doporučuje použití postupu, který vychází z aktuálního času, adresy generující stanice a algoritmu SHA-1. Dokonce je zde výslovný zákaz používání pořadového nebo jinak předvídatelného přidělování. Velikostí této položky a způsobem generování je zajištěna velice malá pravděpodobnost, že si dvě sítě vygenerují stejný identifikátor. Tomu napomáhá také možnost registrace identifikátoru na stránkách organizace SixXS, která provozuje generátor adres. Bohužel, tento registr není globální, a když si někdo vygeneruje identifikátor v jiném generátoru, je toto opatření bezpředmětné.

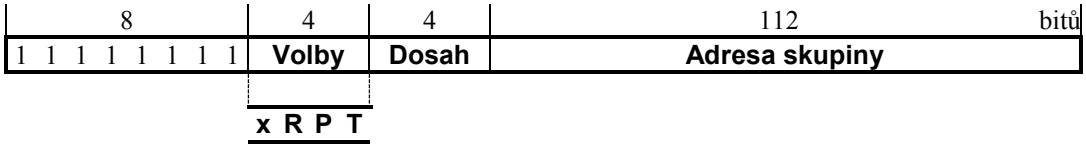
[7] „V pozadí zjevně čeká myšlenka jakési centrální autority, která by u adres s L=0 ručila za celosvětovou jednoznačnost jejich prefixu. Myšlenka globálně koordinovaných lokálních adres má své urputné zastánce i kritiky,“ tvrdí Satrapa ve své knize IPv6 [2].

[8] Lze použít generátor na adrese http://www.sixxs.net/tools/grh/ula/.

Unikátnost těchto adres je velice důležitá. Lze předpokládat, že na jedné páteřní síti, například u jednoho ISP, bude více organizací chtít využít lokálních adres pro své pobočky. Vygenerování unikátního prefixu pro každou z těchto sítí nám zajistí, že nenastanou v páteřní síti žádné kolize v adresách těchto organizací. V případě lokálních místních adres bylo díky konstantnímu prefixu pravděpodobné, že si dvě organizace na jedné páteřní síti určí stejnou podsít a tím hrozilo i velké riziko kolizí mezi těmito adresami. [2], [10]

### 3.2.4 Skupinové adresy

Skupinové adresy nejsou nic nového, už v IPv4 se hojně používají například pro přenosy videa a zvuku v reálném čase. Dokonce by se mezi ně dala zařadit i všesměrová (broadcast) adresa. IPv6 ale přišlo s novou myšlenkou jak tyto adresy inovovat. Byl zaveden koncept dosahu adres, jenž byl využit už u lokálních adres, kde dosah lokálních linkových adres je jen po jedné lince, nebo dosah lokálních místních adres po jednom „místě“. Nejvíce se ale tato vlastnost využije u skupinových adres. Všechny skupinové adresy začínají prefixem FF00::/8, díky kterému je jednoduché je odlišit od ostatních adres. Strukturu dalších 120 bitů názorně ukazuje obrázek 11. [2], [7]




**Obrázek 11 - Struktura skupinové adresy**

*Převzato z [7 str. 13], úpravy autor*

Po základním prefixu následují v adrese 4 bity voleb. Přehled voleb a jejich význam je uveden v tabulce 6. První z voleb je rezervovaná pro pozdější použití a musí být nulová. Dále následují volby R (Rendevous point), P (Prefix) a T (transient). Volba T určuje, zda je adresa skupiny přiřazena trvale nebo dočasně. Trvalé, neboli „dobře známé“ adresy přiděluje organizace IANA. Dočasné adresy se mohou generovat podle potřeby. Kompletní seznam přidělených „dobře známých“ adres lze najít na stánkách organizace IANA[9]. [13]

**Tabulka 6 - Volby skupinových adres**

| Volba | Hodnota | Význam |
|---|---|---|
| **x** | 0 | Rezervovaná volba, musí být nulová |
| **R** | 0 | Nezahrnuje Rendezvous point |
| | 1 | Zahrnuje Rendevous point (musí být T=1 a P=1) |
| **P** | 0 | Nevychází ze síťového prefixu |
| | 1 | Vychází ze síťového prefixu (musí být T=1) |
| **T** | 0 | Dobře známá (trvalá) adresa |
| | 1 | Dočasná adresa |

[9] http://www.iana.org/assignments/ipv6-multicast-addresses/

**Tabulka 7 - Dosahy skupinových adres**

| Dosah | Význam |
|---|---|
| 0 | Rezervováno |
| 1 | Rozhraní (interface); pouze pro jedno rozhraní, použitelné jen pro loopback |
| 2 | Linka (link); jedna síť na fyzické vrstvě, jako Ethernet, WiFi nebo sériové spojení dvou uzlů |
| 3 | Rezervováno |
| 4 | Správa (admin); menší rozsah, který musí být nastaven ručně |
| 5 | Místo (site); jedna síť |
| 6-7 | Nepřiřazeno |
| 8 | Organizace (organisation); několik sítí v jedné organizaci |
| 9-D | Nepřiřazeno |
| E | Globální (global); celý Internet |
| F | Rezervováno |

Další položkou je dosah, což je čtyřbitová položka určující jaká může být vzdálenost jednotlivých členů. Z šestnácti možných rozsahů byly definovány zatím pouze některé. Jejich přehled najdete v tabulce 7. V IPv4 byla problematika dosahu řešena pomocí omezování životnosti paketu (TTL). Nepřiřazené hodnoty můžou definovat správci sítě podle svého uvážení, přičemž by měli dodržet pravidlo, že čím větší číslo, tím větší dosah v Internetu. Například evropské akademické sítě určily, že hodnota „A“ znamená dosah pokrývající celou národní síť. [2]

Skupinová adresa se nikdy nesmí objevit jako adresa odesilatele IPv6 datagramu, ani v rozšiřující směrovací hlavičce.

### 3.2.5 Výběrové adresy

Výběrové adresy jsou úplnou novinkou v IPv6. Základní myšlenka těchto adres je sdružení uzlů poskytujících stejné služby do jedné skupiny s jednou IPv6 adresou. V takovémto případě je zpráva zaslaná na výběrovou adresu doručena nejbližšímu uzlu z této skupiny s využitím standardního směrování.

Výběrové adresy nejsou nijak syntakticky odlišeny od globálních individuálních adres ani pro ně nebyl rezervován žádný adresní prostor, proto je nemožné je od „normálních“ adres rozlišit. Pokud se nastavuje nějakému uzlu výběrová adresa, musí se dbát na správné nastavení. [2], [13]

Pro směrování výběrových adres by mělo být využito standardních způsobů směrování, které zajistí rozšíření informace o nejbližším výběrovém uzlu. To znamená, že pokud není výběrová adresa s místním prefixem, router pro výběrovou adresu musí přidat do své směrovací tabulky zvláštní záznam, což je velkou nevýhodou pro zřízení výběrové skupiny uzlů napříč celým Internetem. Znamenalo by to zvláštní zápis do už tak přeplněných směrovacích tabulek páteřních směrovačů, což oponuje jednomu ze základních principů IPv6 a to snažit se o co nejmenší směrovací tabulky. [2], [7]

Další nevýhodou výběrových adres je, že není možné zajistit, aby dva pakety odeslané na takovouto adresu byly doručeny stejnému uzlu. Tím vzniká velké omezení pro použití výběrových adres a ztrácíme tím možnost použití stavových protokolu.

Výběrové adresování se hodí například pro kořenové DNS servery. S DNS se komunikuje pomocí bezstavového protokolu UDP, u kterého nevadí, že na každý dotaz může odpovědět jiný server. U této služby je třeba zajistit co nejrychlejší odezvu a pokud možno hodně serverů pod malým počtem adres, z důvodu rozkládání zátěže. V neposlední řadě dokáže výběrové adresování pomoci při odolávání útokům o zahlcení (DoS, DDoS) služeb DNS. [2], [13]

# 4 Objevování sousedů

Objevování sousedů je protokol, který umožnuje uzlům na stejné síti oznamovat svoji přítomnost ostatním sousedům a také se z oznámení ostatních učit o svých sousedech v síti. Tento protokol musí podporovat každá implementace protokolu IPv6.

Pomocí této metody jsou nahrazeny protokoly IPv4 jako vyhledání směrovače pomocí ICMP (RDISC), Address Resolution Protocol (ARP) a ICMPv4 přesměrování. Protokol objevování sousedů definovaný v RFC 4861: *Neighbor Discovery for IP version 6 (IPv6)* [14], ale přináší obecnější pojetí problému a je využitelný pro celou řadu dalších nástrojů. Jak uvádí [2], protokol je využíván například pro

- zjišťování linkových adres uzlů ve stejné lokální síti,
- rychlé aktualizace neplatných položek a zjišťování změn v linkových adresách,
- hledání směrovačů,
- přesměrování,
- zjišťování prefixů, parametrů sítě a dalších údajů pro automatickou konfiguraci adresy,
- ověřování dosažitelnosti sousedů,
- detekce duplicitních adres.

Pro svou činnost využívá protokol objevování sousedů 5 standardních typů ICMPv6 zpráv [14]:

- výzva směrovači (Router Solicitation),
- ohlášení směrovače (Router Advertisement),
- výzva sousedovi (Neighbor Solicitation),
- ohlášení souseda (Neighbor Advertisement),
- přesměrování (Redirect).

## 4.1 Hledání linkových adres

Problém zjišťování linkových adres byl již v IPv4, jeho řešením byl protokol ARP, kterému se hledání linkových adres v IPv6 velice podobá. Hlavním rozdílem jsou adresy, na které tazatel odesílá své dotazy. Pro tyto účely byla z adresního prostoru přiřazena část skupinových adres se společným prefixem

FF02:0:0:0:0:1:FF00::/104

Uzel, který pro určitou IPv6 adresu hledá adresu linkovou, vezme posledních 24 bitů z hledané IPv6 adresy a připojí je za daný prefix. Na takto vytvořenou skupinovou adresu uzel zašle svůj dotaz. Například je-li třeba zjistit linkovou adresu pro následující IPv6 adresu:

2001:DB8:1:2:022A:FFF:FE32:5ED1

Uzel pro tuto adresu vygeneruje následující skupinovou adresu:

FF02::1:FF32:5ED1

Tato adresa se odborně nazývá *adresa pro vyzývaný uzel*. Použití části jednoznačného identifikátoru rozhraní má výhodu v tom, že snižuje počet skupin, ve kterých musí být počítač členem. Takto vygenerovaná adresa většinou bude platná pro všechny IPv6 adresy na daném rozhraní. Další výhodou je, že díky dvaceti čtyřbitovému identifikátoru bude v nejčastějších případech ve skupině uzel sám a to s vysokou pravděpodobností i ve velkých sítích. Aby tento systém fungoval, musí uzel při inicializaci síťového rozhraní vstoupit do všech potřebných skupin.

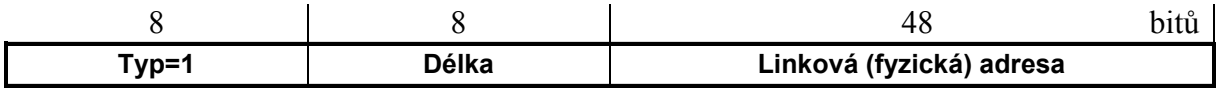
Pokud tedy počítač potřebuje zjistit linkovou adresu cíle, postupuje následovně: Podle způsobu popsaného výše vygeneruje podle IP adresy skupinovou *adresu vyzývaného uzlu*. Poté na tuto adresu pošle ICMP zprávu typu 135, *výzva sousedovi*. Je-li počítač s danou adresou aktivní, odpoví ICMP zprávou typu 136, *ohlášení souseda*, ve které uvede jako volbu svou linkovou adresu. Formát výzvy sousedovi ilustruje obrázek 12. Datagram je velice jednoduchý, v podstatě obsahuje pouze jedinou informaci a to hledanou adresu.

| 8 | 8 | 16 bitů |
|---|---|---|
| **Typ=135** | **Kód=0** | **Kontrolní součet CRC** |
| **Rezervováno** | | |
| **Hledaná adresa** | | |
| **Volby** | **...** | |

**Obrázek 12 - Výzva sousedovi**

*Převzato z [14 str. 22], úpravy autor*

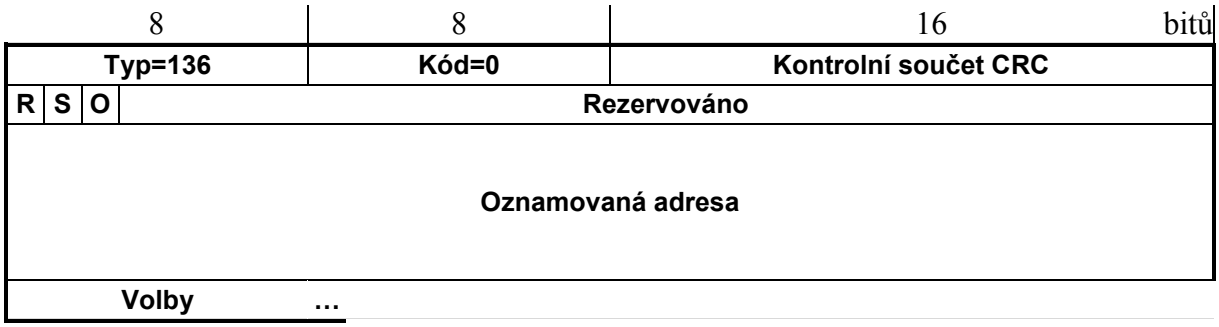
Odesilatel může do výzvy sousedovi přidat volbu s jeho linkovou adresou, aby adresát nemusel pro zjištění jeho linkové adresy znovu odesílat *výzvu sousedovi* a rovnou věděl, kam má odeslat odpověď, *ohlášení souseda*. Formát této volby je uveden na obrázku 13.




**Obrázek 13 - Volba *Linková adresa odesilatele***

*Převzato z [14 str. 28], úpravy autor*

Obrázek 14 ukazuje strukturu zprávy *ohlášení souseda*. Její největší část opět zabírá adresa, tentokrát oznamovaná adresa odesílajícího uzlu. Kromě toho obsahuje tři bity příznaků. Znak R (Router) znamená, že odesilatel zprávy je směrovač. Znak S (Solicited) udává, že tato zpráva byla vyžádána *výzvou sousedovi* a poslední znak O (Override) určuje, jestli tato informace má přepsat případné stávající informace o této adrese.




**Obrázek 14 - Ohlášení souseda**

*Převzato z [14 str. 23], úpravy autor*

Každý síťový uzel by si měl v interní databázi, takzvané *cache sousedů*, uchovávat informace o již zjištěných linkových adresách sousedu v páru s IPv6 adresou.

Podrobnosti k této problematice naleznete v [2] a[14].

# 5 Automatická konfigurace

Jedním z požadavků při vývoji IPv6 bylo ulehčení konfigurace počítačů v síti a co největší zautomatizování této činnosti. Už v IPv4 byla možnost nastavit adresy pomocí DHCP serveru, ale IPv6 přišlo i s možností automatické konfigurace hostů bez jakéhokoliv serveru. IPv6 nabízí tedy dva typy automatického přidělování adres:

- Stavová konfigurace – funguje na stejném principu jako DHCP v IPv4.
- Bezstavová konfigurace – pomocí zpráv zvaných *ohlášení routeru*.

## 5.1 Stavová konfigurace

Model stavové konfigurace adres se používá už řadu let – od RARP, před BOOTP až k dnešnímu DHCP. Pro využití s IPv6 byl vyvinut protokol DHCPv6, jehož specifikace byla uvedena v dokumentu RFC 3315: *Dynamic Host Configuration Protocol for IPv6 (DHCPv6)* [15].

V tomto dokumentu jsou definováni tři účastníci DHCP. *Server*, který poskytuje informace, *zprostředkovatel*, který zprostředkovává spojení mezi serverem a klientem pokud se nacházejí na jiných linkách a nakonec *klient*, který informace od serveru žádá. Server a zprostředkovatel se většinou shrnují pod jeden název, *agent*. Obecně se za agenta dá považovat zařízení, které sídlí na stejné lince jako klient a poskytne mu DHCP odpověď, ať už jde o odpověď zprostředkovanou, či vlastní. [15]

DHCPv6 přichází s novým způsobem identifikace hostů a serverů. V DHCPv4 se k tomuto účelu používaly MAC adresy síťových rozhraní, ale DHCPv6 zavádí takzvané DUID. Hlavní myšlenkou DUID je přiřazení jednoznačného identifikátoru určitému počítači po celou dobu jeho života a ne jen různé identifikátory síťovým rozhraním. Tento identifikátor by měl být pokud možno stálý a neměnit se ani při výměně síťové karty. [2], [16]

Vývojáři definovaly v RFC 3315 [15] tři způsoby generování DUID:

- Vytvoření výrobcem a přiřazení k určitému PC.
- Generování pomocí kombinace linkové adresy a času vytvoření. Tento způsob spoléhá na možnost uložení identifikátoru do trvalé paměti.
- Samotná linková adresa. Zvolí se jedna linková adresa a ta reprezentuje celé PC. Tento identifikátor nevyhovuje požadavku stálosti.

Protokol DHCPv6 využívá ještě jeden druh identifikačních konstrukcí a to jsou takzvané identifikační asociace (IA). IA se skládá z konfiguračních informací pro jednotlivá rozhraní. Každé takovéto rozhraní je ještě označeno jednoznačným identifikátorem IAID, který by měl být také stálý, nejspíš tedy uložený v trvalé paměti nebo generován algoritmem, který bude mít v různém čase stejné výsledky. Protokol

DHCPv6 tedy pro svou činnost využívá dva druhy identifikátorů. Pomocí DUID je jednoznačně identifikován klient a pomoci IAID klientovo rozhraní. [2], [15]

Pokud chce například nově připojený klient do sítě získat adresu z DHCP serveru. Nejprve klient odešle *výzvu* na skupinovou adresu všech DHCP agentů na lince (FF02::1:2), do které připojí své DUID a všechna IA. Pokud *výzvu* obdrží server, odpoví klientovi *ohlášením serveru*, ve kterém nabídne konfigurační parametry, které by přidělil jednotlivým IA. Pokud *výzva* dorazí ke zprostředkovateli, ten zabalí dotaz do nové zprávy typu *předání* a odešle ji na adresy definované v jeho seznam DHCP serverů (takovou adresou může být i skupinová adresa všech DHCP serverů místa FF05::1:3). Server pak odpoví zprostředkovateli zprávou zabalenou jako *zprostředkovaná odpověď*, ten ji rozbalí a odešle zpět ke klientovi. Přehled všech typů zpráv využívaných v komunikaci DHCPv6 je uveden v tabulce 8. [2], [15]

**Tabulka 8 - Typy zpráv DHCPv6**

| Typ | Význam |
|---|---|
| 1 | Výzva (solicit) |
| 2 | Ohlášení serveru (advertise) |
| 3 | Žádost (request) |
| 4 | Potvrzení (confirm) |
| 5 | Obnovení (renew) |
| 6 | Převázání (rebind) |
| 7 | Odpověď (reply) |
| 8 | Uvolnění (release) |
| 9 | Odmítnutí (decline) |
| 10 | Rekonfigurace (reconfigure) |
| 11 | Žádost o informace (information request) |
| 12 | Předání (relay forward) |
| 13 | Zprostředkovaná odpověď (relay reply) |

Klient shromáždí všechny nabídky, vytvoří si tak seznam všech dostupných DHCP serverů a vybere si jeden z nich (měl by ten s největší preferencí), kterému pošle *žádost* o přidělení adres. V žádosti uvede DUID serveru, kterému je určena, ale zprávu opět pošle na adresu všech DHCP agentů v síti. Servery, kterým tato žádost nepatří, ji ignorují. Cílový server žádost vyhodnotí a odešle nazpět *odpověď*, ve které oznámí klientovi adresy. Klient si přidělené adresy ověří, a pokud zjistí, že je už někdo používá, může je odmítnout a informovat o tom DHCP server pomocí zprávy *odmítnutí*. [2]

## 5.2 Bezstavová konfigurace

Tento způsob konfigurace umožňuje, aby si klient mohl nastavit adresu bez jakéhokoliv serveru. Stačí mu k tomu získat takzvané *ohlášení směrovače*, které routery vysílají v náhodném intervalu do všech sítí, ke kterým jsou připojeny. Tím se všechny počítače dozvědí, v jaké jsou sítí, jak se tu komunikuje a kdo je implicitní směrovač. *Ohlášení směrovače* obsahuje všechny prefixy adres, které se v dané síti používají.

Postup nastavení adres pro nově připojený uzel v sítí je následující: Nejprve si daný uzel vygeneruje lokální linkovou adresu pro své rozhraní připojením identifikátoru svého rozhraní k prefixu FF80::/10. Pro jistotu ověří, zda již v sítí někdo stejnou adresu nepoužívá, což je velmi málo pravděpodobné. Dále vyčká na ohlášení směrovače, případně o něj může aktivně požádat pomocí *výzvy směrovači*.

Z ohlášení směrovače se dozví, jaké se v síti používají prefixy a zda je u nich možné použít bezstavovou konfiguraci. Pokud ano, tyto prefixy vezme, připojí za ně identifikátor svého rozhraní a takto vytvořené adresy přidělí k danému rozhraní. [2], [16], [17]

Další potřebné údaje lze doplnit pomocí bezstavového DHCPv6, jehož definici obsahuje dokument RFC 3736: *Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6* [18].

# 6 Směrování, směrovací protokoly

V IPv6 sítí musí každé zařízení umět směrovat, i přes to, že specialisté na směrování jsou směrovače. Základem směrování je směrovací tabulka, ve které se uchovává informace o tom, kudy se dá k danému cíli dostat. V tabulce se ukládají IPv6 prefixy a jejich délka. V případě, že je cíl připojen k zařízení přímo, je v tabulce uvedeno rozhraní, přes které je připojen. Jinak je zde uvedena IPv6 adresa dalšího uzlu na cestě za cílem. Speciálním prefixem, který se ve většině tabulek nachází je „::/0“. Tento prefix udává implicitní cestu k cílům, které v tabulce nemají žádnou jinou vyhovující hodnotu.

Princip tohoto základního směrování se od IPv4 vůbec nezměnil. Když stroj potřebuje předat datagram, vyhledá ve své směrovací tabulce všechny prefixy, které odpovídají cílové adrese datagramu. Těch může být několik, protože například implicitní cesta bude vyhovovat všem adresám. Z takto vybraných prefixů vybere ten nejdelší a předá datagram podle údajů uvedených u tohoto prefixu.

Toto základní směrování musí podporovat všechna zařízení používající IPv6. [2], [3]

## 6.1 Směrovací protokoly

Směrování na koncových stanicích je většinou velice jednoduché. Směrovací tabulka obsahuje jenom pár záznamů a většinou se ani moc nemění. Mnohem těžší to mají směrovače, které propojují několik sítí. Podle počtu sítí se také odvíjí velikost směrovací tabulky. Dalším požadavkem na směrovače je dynamičnost a přizpůsobování se změnám v prostředí. Proto se používají různé směrovací protokoly, pomocí kterých se jednotlivé směrovače informují o připojených sítích a upravují si své tabulky.

Páteřní směrovače mají tuto úlohu nejtěžší, protože musí propojit nejvíce sítí. Proto se využívá hierarchické struktury adresování, která pomáhá snižovat počty záznamů v páteřních směrovačích pomocí shlukování sítí do kratších prefixů.

Aby bylo vůbec možné všechny tyto směrovače obstarat, rozdělují se do takzvaných autonomních systémů (AS). Autonomní systém je tvořen směrovači, které mají jednotnou správu a směrovací politiku. Například jeden autonomní systém muže být síť jednoho poskytovatele Internetu a sítě jeho zákazníků. V důsledku tohoto rozdělení jsou i směrovací protokoly rozděleny do určitých skupin, IGP a BGP. [2], [3]

### 6.1.1 IGP

Jako IGP, neboli *Internal Gateway Protocol*, se označují směrovací protokoly sloužící ke směrování v jednom autonomním systému. Takovéto protokoly kladou důraz hlavně na rychlou reakci na změny v síti, většinou se v jednom AS nenachází astronomická množství sítí, takže i jejich směrovací tabulky nebudou extrémně veliké. V současné době jsou tři IGP směrovací protokoly, které podporují IPv6 a to RIPng, OSPFv3, IS-IS a EIGRP. [2]

### 6.1.2 EGP

Naproti tomu, k výměně směrovacích informací mezi autonomními systémy slouží *External Gateway Protocol*. Prostřednictvím těchto protokolů se směrovače dozvídají o cestách kudy do kterého autonomního systému a jaké v něm jsou prefixy. Tyto protokoly spojují všechny sítě v jeden celek, Internet. Hraniční směrovače autonomních systémů tedy musí obsahovat ve směrovací tabulce informace o cestách do celého Internetu, jejich tabulky tímto nabývají na objemu a je třeba, aby na to byly připraveny.

Protokoly z této skupiny se zaměřují hlavně na to, aby toto ohromné množství informací zvládly, proto také jejich reakce na změny v topologii je pomalejší. V současné době se využívá pouze jeden EGP protokol a to BGP4. Pro využití s IPv6 byla vyvinuta upravená verze BGP4+. [2]

### 6.1.3 RIPng

RIP, čili Router Information Protocol, je jeden z nejstarších směrovacích protokolů řadících se do kategorie IGP. Pořád se ale drží mezi používanými, hlavně díky své jednoduchosti a ani omezení, která jsou spojena s jeho užíváním, to neovlivní. Jedním z problémů je například omezená maximální délka cesty na 15 skoků nebo pomalejší reakce na změny. Pro menší sítě je ale nejjednodušší volbou. RIPng, neboli RIP nové generace vychází z RIPu verze 2 a liší se od něj v zásadě jen v podpoře IPv6 adres. Specifikaci tohoto protokolu byste našli v RFC 2080: *RIPng for IPv6* [19].

Tento protokol funguje na principu vektoru vzdáleností. Pro každou síť si uchovává „cenu“ spojení, která je definována součtem cen linek, přes které musí datagram projít. RIPng se snaží směrovat datagramy k cíli vždy tou cestou, která má nejmenší „cenu“.

Každých 30 vteřin[10] RIPng odesílá údaje ze své směrovací tabulky, aby informoval o situaci okolní směrovače. Když sám obdrží tabulku od některého ze sousedů, přičte k ní cenu spojení, ze kterého přišla a porovná ji se svou tabulkou, pokud zde nalezne nějakou výhodnější cestu, novou cestu, nebo se nějaká cesta přes tento spoj zhoršila, upraví podle toho svou tabulku.

RIP je určen pro menší sítě. Tomu také odpovídá rozmezí hodnot použitelných pro ohodnocení cesty. Povolené hodnoty jsou 1-15 a hodnota 16 označuje nedosažitelnou síť. V praxi to pak vypadá tak, že každá linka má cenu 1.

RIPng posílá svou směrovací tabulku v následujících případech [2]:

- *Pravidelné aktualizace* zasílané každých 30 vteřin s náhodným rozptylem od -15 do 15 vteřin, aby nedocházelo k nežádoucí synchronizaci mezi zprávami jednotlivých směrovačů.

[10] S náhodným rozptylem od -15 do 15 vteřin.

- *Aktualizace vyvolané změnou* směrovač zasílá, když došlo ke změně v jeho směrovacích tabulkách.

- *Odpověď na požadavek* směrovač odesílá, když některý ze sousedů o zaslání informací zažádal.

První dva typy zpráv směrovač odesílá všem sousedům, což znamená do všech sítí, ke kterým je připojen, na skupinovou adresu všech RIPng směrovačů (FF02::9). Odpověď na požadavek zasílá pouze žadateli. [2], [3], [20]

### 6.1.4 OSPFv3

Dalším protokolem z řady IGP směrovacích protokolů je *Open Shortest Path First* (OSPF). Tento protokol je mnohem mladší než RIP, je také o dost složitější a sofistikovanější. OSPF pracuje na principu stavu linek. To znamená, že si každý směrovač buduje a uchovává mapu sítě, která obsahuje informace o tom, jak jsou jednotlivé směrovače mezi sebou propojeny, jaké mají prefixy podsítí, ceny linek a mnoho dalších informací. [2], [3]

Z této mapy každý směrovač vypočítá strom nejkratších vzdáleností ke všem známým cílům, jehož vrcholem je směrovač sám. Touto metodou OSPF zjistí, kudy vede nejkratší cesta ke všem cílům a tu zavede do standardní směrovací tabulky. Proto je důležité, aby všechny směrovače v síti měli stejnou mapu sítě. To je zajištěno zprávami, které směrovač odesílá okamžitě při jakékoliv změně v této mapě. Výhodou tohoto algoritmu je možnost zajistit rychlou odezvu na změny v síti a také schopnost zvládnout i rozsáhlejší sítě. [2]

V tabulce 9 jsou uvedeny typy zpráv, kterými mezi sebou jednotlivé směrovače komunikují. Tyto zprávy se v OSPF nazývají *Link State Advertisement* (LSA). Pomocí těchto zpráv si směrovače budují mapy sítě.

**Tabulka 9 - Typy LSA zpráv**

| Kód | Název | Význam |
|---|---|---|
| 1 | směrovač | Stav rozhraní daného směrovače, posílá každý směrovač |
| 2 | Síť | Seznam směrovačů připojených k síti, posílá pověřený směrovač |
| 3, 4 | Souhrn | Cesta k cíli, jenž leží mimo danou oblast, ale uvnitř autonomního systému, posílají hraniční směrovače oblasti |
| 5 | Externí | Cesta k cíli z jiného autonomního systému, posílají hraniční směrovače autonomních systémů |

Sousedící směrovače musí neustále udržovat synchronní mapu sítě. Vždy při vytvoření sousedství mezi směrovači si směrovače vymění několik zpráv *Popis databáze*, ve kterých uvedou všechny LSA a jejich verze, které znají. Protějšek si poznamená

všechny LSA, které dosud neznal, nebo znal jejich starší verzi. Poté o ně požádá pomocí *Žádosti o stav linky* a očekává *Aktualizaci stavu linky* s požadovanými údaji. Po dokončení této výměny oba směrovače vědí, že jejich mapy jsou totožné. Ve chvíli kdy dojde k jakékoliv změně v mapě jakéhokoliv směrovače, tento směrovač ihned odešle *Aktualizaci stavu linky* a tím o změně informuje všechny sousední směrovače. Díky těmto metodám je reakce na změny v síti velmi rychlá a navíc jsou odesílány pouze změny a tím je hodně ušetřeno na režii. [2]

Tento směrovací protokol má další zajímavé vlastnosti, například je připraven na zvládnutí opravdu velkých sítí pomocí rozdělení této sítě na oblasti, ve kterých budou směrovače uchovávat mapy pouze pro danou oblast. Dále nabízí možnost zabezpečení komunikace mezi směrovači pomocí IPsec. OSPFv3 je ve své podstatě stejné jako OSPFv2 pro IPv4, čímž i ve verzi pro IPv6 zůstali například identifikátory routeru v podobě IPv4 adres. [2]

### 6.1.5 BGP4+

*Border Gateway Protocol* (BGP) je jediných zástupcem skupiny EGP protokolů. Dalo by se říct, že tento protokol drží celý Internet pohromadě. Jako předchozí protokoly vychází ze směrovacího protokolu pro IPv4 (BGP), který po úpravách uvedených v dokumentu RFC 4760: *Multiprotocol Extensions for BGP-4* [21] byl rozšířen tak, že podporuje směrování prakticky jakéhokoliv síťového protokolu, tedy i IPv6. Tato upravená verze se označuje jako BGP4+.

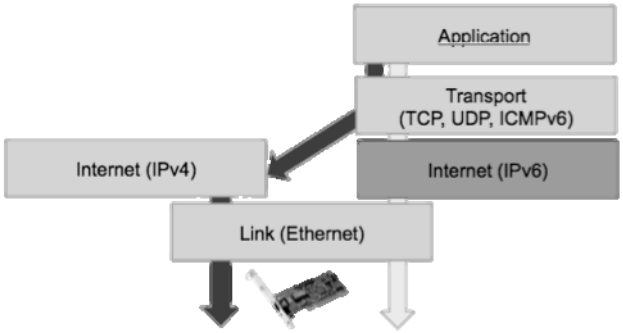
BGP není tak dynamické jako předchozí IGP protokoly. Klade důraz hlavně na rozšiřitelnost a zvládnutí ohromného množství směrovacích informací, dle svého algoritmu nemusí vždy vybrat nejoptimálnější cestu [22]. Dále své sousedy si nevyhledává sám, ale musí je zadat správce sítě [2]. Jedním z dalších rozdílů je, že sousední směrovač nemusí být ve stejné podsíti jako u IGP. Pro komunikaci se svými sousedy používá TCP protokol, podle kterého také zjišťuje, zda jsou jeho sousedé dosažitelní.

# 7 Přechod z IPv4 na IPv6

Jedním ze zásadních problémů přechodu na nový komunikační protokol je to, že nejde uskutečnit ze dne na den a nějaký čas spolu musí tyto dva protokoly koexistovat. Kvůli tomu bylo vyvinuto několik přechodových mechanizmů, které nám usnadní toto překlenovací období zvládnout.

## 7.1 Dvojí zásobník (Dual Stack)

Tato metoda není zas tak úplně přechodový mechanizmus. Principem je provozování jak IPv4 tak i IPv6 konektivity na jednom stroji. Název, který tento způsob spojení se světem dostal je trochu zavádějící. Podle názvu by se mohlo zdát, že zařízení má pro každý protokol zvláštní zásobník a vyřizuje požadavky zcela odděleně. Není tomu tak, ve většině implementací je totiž jeden hybridní zásobník a požadavky se vyřizují postupně. [2]




**Obrázek 15 - Princip Dual Stack**

*Převzato z Lupa.cz [23]*

Tento způsob je asi „nejčistším“ a nejspolehlivějším, ale zároveň nejnáročnějším, připojením do IPv4 i IPv6 sítě zároveň. Je třeba vše konfigurovat dvakrát, počínaje adresami pro jednotlivé uzly, přes různé firewally a jiné služby v síti. V podstatě se musí provozovat dvě logické sítě na jedné fyzické infrastruktuře. [23]

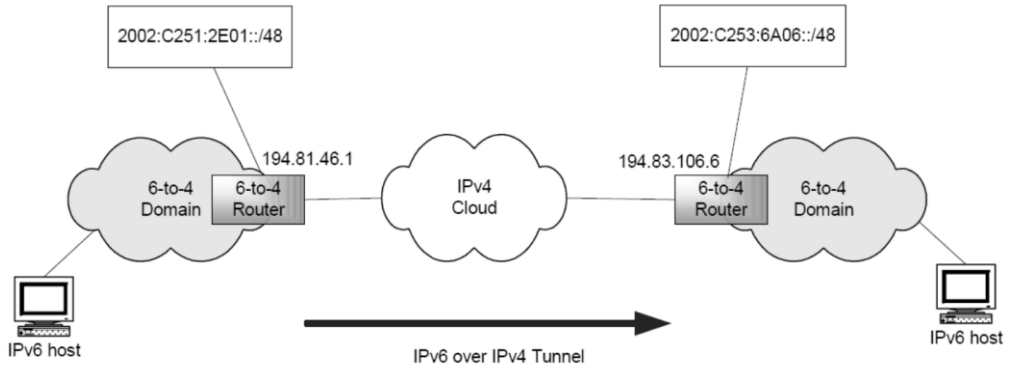
## 7.2 Tunelování

Tunelování není ve světě počítačových sítí žádnou novinkou. V současné době se používá například k propojení několika geograficky oddělených sítí jedné společnosti. VPN a šifrovaná tunelovaná spojení třeba pomocí SSH jsou dalšími příklady.

Cílem jakéhokoliv tunelování je snaha překonat nepřátelské území, v našem případě je to IPv4 síť. Prakticky jde o to, že stroj zahajující tunelové spojení zabalí to, co je třeba přenést (pro nás je to IPv6 paket) jako data do paketu IPv4. Když dorazí paket na konec tunelu, stroj rozbalí IPv4 paket a získá původní IPv6, se kterým pak zachází podle standardních směrovacích pravidel, pokud tedy není paket adresován přímo jemu. [2], [23]

### 7.2.1 6to4

Jedním z nejpoužívanějších přechodových mechanizmů na bázi tunelování je protokol 6to4. Jeho definici můžete najít v RFC 3056: *Connection of IPv6 Domains via IPv4 Clouds*. Tento protokol je určen pro spojení IPv6 sítí mezi sebou přes IPv4 síť, tudíž jde o tunelové spojení mezi dvěma směrovači. Tyto směrovače musí mít přiřazenou vlastní veřejnou IPv4 adresu, aby mohli komunikovat po IPv4 síti. Pro potřeby 6to4 protokolu byl organizací IANA přidělen prefix 2002::/16. Každá izolovaná IPv6 síť, která chce využívat protokol 6to4, musí mít přiřazen prefix 2002:*IPv4_adresa*::/48, kde *IPv4_adresa* je veřejná adresa 6to4 směrovače [3]. V dalších 16 bitech má možnost správce sítě vytvořit subnety a takto vytvořený prefix přiřadit všem uzlům v síti. Názornou ukázku propojení 6to4 sítí najdete níže na obrázku 16. Pomocí 6to4 nejdou propojit jen směrovače, ale i host se směrovačem, podmínkou jsou ale veřejné adresy zúčastněných. [24]




**Obrázek 16 - Komunikace mezi 6to4 sítěmi**

*Převzato z IPv6 Deployment Guide [3 str. 64]*

Tento protokol využívá automatického spojování tunelů, což přidává na jeho jednoduchosti nasazení. Není třeba žádných úprav v IPv4 směrování a v IPv6 akorát nahlásit výchozí cestu do 6to4. Když na takovýto směrovač dorazí paket se 6to4 prefixem, směrovač vezme z cílové adresy třetí až šestý bajt, ve kterém se nachází adresa cílového 6to4 směrovače. Poté s tímto směrovačem automaticky vytvoří tunel, IPv6 paket zabalí do IPv4, kde jako protokol uvede 41 (IPv6) a data odešle na IPv4 adresu získanou z cílové adresy IPv6 paketu. [2], [23]

Problém nastává, když bude třeba spojení s nativní IPv6 sítí, k tomu 6to4 musí využít nějaký relay server, který má konektivitu do IPv4 i do nativní IPv6 sítě. K takovémuto relay serveru je těžké se vůbec dostat. Někteří ISP provozují své vlastní relay servery, ale ty nabízejí jen ve své síti a veřejných relay serverů není mnoho [3]. Dále komunikace tam nemusí jít stejnou cestou zpět a po cestě se mohou některé pakety ztratit, tento problém platí i pro spojení mezi dvěma 6to4 sítěmi.

Tento protokol má také mnoho problémů, které se časem objevili a již existuje dlouho diskutovaný návrh [25] na označení tohoto protokolu za historický. V tomto návrhu jsou uvedeny tyto důvody pro zamítnutí 6to4 protokolu:

- Využívání relay serverů třetích stran pro spojení s nativní IPv6 sítí.
- Umístění relay serverů může značně zvyšovat odezvu.
- Relay servery lze jednoduše použít k anonymizování komunikace a využít je pro různé útoky.
- 6to4 může skrytě ztrácet pakety, kvůli firewallům kde je zakázán pro ně neznámý protokol 41.
- Konce 6to4 tunelů vždy musí mít veřejnou IPv4 adresu.

### 7.2.2 6over4

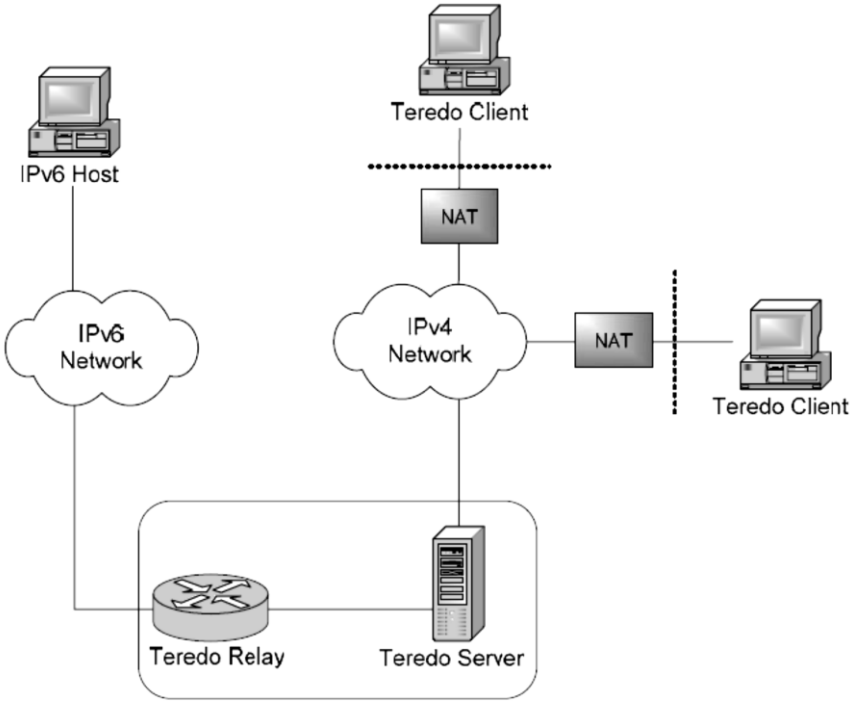
Protokol 6over4 definovaný v RFC 2529: *Transmission of IPv6 over IPv4 Domains without Explicit Tunnels* měl sloužit k propojení odloučených IPv6 hostů s IPv6 světem. Tento mechanizmus v podstatě využívá IPv4 jako linkovou přenosovou vrstvu. Ke své funkčnosti potřebuje podporu multicastu v IPv4, protože používá standardní mechanizmy IPv6 jako například objevování sousedů a automatickou konfiguraci. IPv6 skupinové adresy si mapuje do IPv4 prefixu 239.192.x.y, kde x a y jsou poslední dva bajty IPv6 skupinové adresy [2]. Potřeba podpory skupinového adresovaní v IPv4 sítí značně tento protokol omezila a proto se v praxi dočkal velice málo implementací a některé knihy ho dokonce doporučují nepoužívat. [3]

### 7.2.3 Teredo

Teredo je jeden z dalších protokolů umožňující propojení IPv4 a IPv6 pomocí automatických tunelů. Definice tohoto protokolu se nachází v RFC 4380: *Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs)*. Teredo se zaměřuje především na propojení hostů s neveřejnou IP adresou skrytou za NATem. Uzly, které chtějí využívat Teredo musí samozřejmě podporovat oba síťové protokoly, jak IPv4, tak IPv6.

Tento protokol pro své potřeby využívá Teredo server a Teredo relay. Teredo server se použije při každé komunikaci klienta. Nejdříve klient potřebuje získat od serveru IPv6 adresy a zaregistrovat se. Učiní tak odesláním IPv6 *Výzvy směrovači* zabalené do IPv4 paketu na Teredo server. Server klientovi odpoví *Ohlášením směrovače* s 64bitovým prefixem složeným z globálního Teredo prefixu (2001::/32) a IPv4 adresy směrovače. Podle normální auto konfigurace IPv6 adresy by si měl klient k tomuto prefixu připojit jedinečný identifikátor rozhraní, ale v případě Tereda tomu tak není. Druhá polovina adresy se skládá z příznaků (2 bajty), nejdůležitější je nejvyšší bit, který určuje, zda je NAT v klientovi síti trychtýřový, či nikoliv. Následuje UDP port použitý v NATu a veřejná IPv4 adresa NATu. Tyto dvě poslední položky jsou zamaskovány invertováním každého jejich bitu [2], [3]. Teredo servery se musí konfigurovat manuálně

a s největší pravděpodobností už je nakonfiguroval výrobce OS. Například v každém OS od Microsoftu, který podporuje Teredo bude nakonfigurován server teredo.ipv6.microsoft.com [24]. Na obrázku 17 je vidět možné zapojení Teredo komponent v síti.




**Obrázek 17 - Struktura sítě s protokolem Teredo**

*Převzato z IPv6 Deployment Guide [3 str. 66]*

Komunikace mezi Teredo klienty probíhá přímo, vytvořenými „dírami“ v NATech. Ale v případě, že klient potřebuje komunikovat se strojem v jiné než Teredo sítí musí se spojit přes takzvaný Teredo relay. Zde je ale největší kámen úrazu celého Tereda a to špatná odezva v komunikaci. Při požadavku na takovouto komunikaci je totiž vyhledán relay server, který se nachází nejblíž Teredo serveru a ne klientovi, tudíž cesta k cíli je většinou zbytečně delší [26]. Další překážkou pro Teredo je symetrický NAT. Microsoft ale tvrdí, že pokud je jen jeden z klientů za symetrickým NATem, Windows Vista si s tím poradí. [24]

## 7.3 Překládání

Překládání je metoda potřebná pro komunikaci zařízení, která na jednom konci podporují pouze IPv6 a na druhém pouze IPv4. Zařízení na tomto principu jsou v IPv4 hojně využívána jako dočasné řešení nedostatku IPv4 adres. Dokáží za jednu IPv4 adresu skrýt například domácí nebo podnikovou síť.

### 7.3.1 NAT-PT

NAT-PT, neboli Network Address Translation – Protocol Translation je vlastně nástupce NATu na IPv4, akorát nepřekládá neveřejné IPv4 adresy na veřejné IPv4, ale

překládá IPv6 na IPv4. Stejně jako u IPv4 může NAT-PT mapovat adresy na několik vybraných IPv4 adres nebo s použitím mapování portů lze překládat pouze na jednu IPv4 adresu, takovémuto mechanizmu se správně říká NAPT-PT, ale v praxi se toto označení nepoužívá a zůstává se u názvu NAT-PT. NAT-PT je definován v RFC 2766: *Network Address Translation - Protocol Translation (NAT-PT)*.

Tento mechanizmus se snaží zajistit komunikaci jak z IPv4 do IPv6 tak obráceně a při tom vzniká zásadní problém této metody a to jsou DNS záznamy, aby byly počítače ve vnitřní síti dostupné z vnějšku je potřeba jejich zavedení do DNS. V takovémto případě je třeba, aby autoritativní DNS server pro tuto síť byl umístěn v natované síti a všechny požadavky procházely přes překladač. Ten musí takovéto dotazy měnit z typu A na AAAA a pak při odpovědi zas obráceně, zároveň si musí vytvořit záznam s mapováním pro příchozí spojení. V opačném směru je to s DNS záznamy ještě složitější. Zejména z tohoto důvodu vydalo IETF dokument RFC 4966: *Reasons to Move the Network Address Translator - Protocol Translator (NAT-PT) to Historic Status*, ve kterém označuje NAT-PT za historický. [2], [3]

### 7.3.2 NAT64

Protože pro tyto účely překládání adres je NAT jedinou rozumnou možností, vznikl po zkušenostech s NAT-PT protokol nový a to NAT64. Je to ještě horká novinka uvedená v RFC 6146: *Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers* v dubnu 2011.

NAT64 vychází z NAT-PT a poučuje se z jeho chyb. Hlavním rozdílem je možnost zahájení komunikace pouze z IPv6 sítě do IPv4 a označování pozměněných DNS dotazů. Pro mapování IPv4 adres v lokální sítí NAT64 využívá 96bitový prefix, za který se pouze připojí IPv4 adresa, toto mapování je statické. Pokud se v síti nacházejí dva NATy, pro každý se zvolí jiný prefix [2]. Použitím NATu v sítí ztrácíme spoustu výhod, které IPv6 přináší, ale máme možnost připojit se k IPv4 sítí například pomocí jedné IPv4 adresy.

NAT64 spolupracuje se službou DNS64, která poskytuje počítačům v natované sítí získat IPv6 adresu serverů, které jsou dostupné na IPv4. Klient pak začne navazovat spojení na IPv6 adresu, kterou NAT64 přeloží na správnou IPv4 adresu. [23]

# 8 Podpora v zařízeních

Podpora v současných operačních systémech a „lepších“ síťových zařízení je na celkem dobré úrovni. Na podporu IPv6 vzniklo sdružení IPv6 Fórum, které mimo své osvěty, vytvořilo certifikační program IPv6 Ready, do kterého se mohou výrobci a vývojáři přihlásit a nechat otestovat své výrobky. Na podrobný seznam zařízení a software se můžete podívat na internetových stránkách IPv6 Ready[11]. [2]

## 8.1 Konfigurace na MS Windows

Podpora pro IPv6 v operačních systémech společnosti Microsoft je již od verze Windows XP SP 1. Postupem času se podpora v jednotlivých verzích zlepšuje, od Windows Vista je k dispozici i základní grafické rozhraní pro nastavení adres IPv6, se kterým si vystačí leckterý běžný uživatel. A ty ještě běžnější si tohoto nastavení ani nevšimnou, vše se nakonfiguruje automaticky a budou připojeni i k IPv6. Ve Windows Vista a Windows 7 je totiž podpora IPv6 ve výchozím stavu zapnuta. Pokud chceme pokročilejší možnosti konfigurace IPv6 protokolu musíme se smířit s tím, že se neobejdeme bez příkazového řádku. Všechny možnosti IPv6 je možné nastavit pomocí příkazu `netsh`. Tento program lze na konfiguraci IPv6 využít ve všech verzích Windows podporujících IPv6. Dále se podíváme na některé možnosti nastavení pomocí tohoto příkazu.

Následujícím příkazem je možné zobrazit výpis všech IPv6 rozhraní v systému.

```
netsh interface ipv6 show interface
```

Pro výpis všech přidělených adres můžeme použít příkaz:

```
netsh interface ipv6 show addresses
```

Nová adresa lze přiřadit rozhraní takto:

```
netsh interface ipv6 add address jméno_rozhraní adresa
```

Dalším příkazem lze vytvořit manuální 6to4 tunel, poté je potřeba tunelu přiřadit IPv6 adresu pomocí předcházejícího příkazu.

```
netsh interface ipv6 add v6v4tunnel jméno místní_IPv4 cílová_IPv4
```

K zobrazení nastavení DNS serverů slouží tento příkaz:

```
netsh interface ipv6 show dnsservers
```

Pokud bychom chtěli DNS server přidat, uděláme to následovně:

```
netsh interface ipv6 add dns adresa_serveru
```

Směrovací tabulka pro IPv6 lze zobrazit tímto příkazem:

```
netsh interface ipv6 show route
```

Záznam do směrovací tabulky přidáme následovně:

```
netsh interface ipv6 add route cílová_adresa jméno_rozhraní
```

[11] https://www.ipv6ready.org/db/index.php/public/

Pro zobrazení sousedů v síti slouží příkaz:

```
netsh interface ipv6 show neighbors
```

Přechodové mechanizmy, které jsou ve Windows implementovány, jsou nastaveny tak, abychom se nemuseli o nic starat. Například tunelování pomocí 6to4 se nastaví automaticky. Vytvoří se rozhraní, nastaví se adresa a vyhledá zprostředkovatel 6to4.ipv6.microsoft.com. Stejně tak Teredo se nastaví kompletně samo. Spojí se s defaultně nastaveným Teredo serverem teredo.ipv6.microsoft.com a je připraveno.

Další možnosti nastavení se můžete dočíst ve zdrojích [2], [24] a [27].

## 8.2 Konfigurace Cisco směrovače

Asi nejdůležitějším síťovým prvkem jsou směrovače, bez nich bychom mohli mít jen mnoho izolovaných sítí a nepropojili bychom je. V této podkapitole se zaměřím na základní konfiguraci na směrovačích jedné z nejvýznamnějších síťových firem, Cisco Systems.

Zásadním příkazem pro povolení směrování IPv6 paketů je

```
ipv6 unicast-routing
```

Na rozhraní aktivujete IPv6 pomocí příkazu

```
ipv6 enable
```

Pro nastavení adres použijeme v konfiguračním módu rozhraní příkazy

```
ipv6 address adresa/délka_prefixu
ipv6 address prefix/délka_prefixu eui-64
```

První příkaz přidělí rozhraní přesně zadanou adresu a druhý zadá pouze prefix, ke kterému si směrovač automaticky doplní identifikátor rozhraní EUI-64.

Následující příkaz v privilegovaném módu zobrazí výpis IPv6 rozhraní s adresami

```
show ipv6 interface
```

Ohlášení směrovače s prefixy přiřazenými rozhraním pro automatickou konfiguraci klientů rozesílá směrovač automaticky, pokud bychom chtěli toto chování ovlivnit, můžeme k tomu použít následující příkazy:

```
ipv6 nd prefix prefix/délka
ipv6 nd managed-config-flag
ipv6 nd other-config-flag
```

Prvním příkazem lze ručně nastavit, jaké prefixy mají být pomocí ohlášení směrovače šířeny, pokud nějaký prefix nastavíme, zruší se implicitní odesílání prefixu rozhraní. Druhý příkaz přiřadí do ohlášení směrovače příznak o použití DHCPv6 serveru a třetí příkaz o použití bezstavového DHCPv6.

Pomocí dalších příkazů lze nastavit tunel pro přechodový mechanizmus 6to4. Nejprve je třeba vytvořit virtuální rozhraní pro tunel. Přiřadit mu globální individuální adresu. Zvolit mód tunelu na 6to4 a určit zdrojové rozhraní pro tento tunel.

```
interface tunnel1
     ipv6 address adresa/prefix
     tunnel mode ipv6ip 6to4
     tunnel source FastEthernet 0/0
```

Další z podporovaných přechodových mechanizmů je překladač NAT-PT. Jeho konfiguraci si ukážeme v pár dalších příkazech.

U každého rozhraní, které se bude překladu účastnit, ať už je vnitřní nebo vnější, je třeba NAT zapnout pomocí příkazu

```
ipv6 nat
```

U IPv4 rozhraní je třeba nastavit prefix pro mapování IPv4 adres, to provedeme příkazem

```
ipv6 nat prefix prefix/délka
```

Dále je nutné nastavit ACL, podle kterého bude překladač usuzovat, jaké vnitřní adresy může překládat a které ne.

```
ipv6 access-list natptACL
     permit ipv6 prefix/délka_prefixu any
```

Nakonec máme dvě možnosti, pokud máme k dispozici více veřejných IP adres, můžeme je přiřadit do takzvaného poolu a umožnit NATu překládat na všechny tyto adresy.

```
ipv6 nat v6v4 pool název_poolu první_IPv4 poslední_IPv4 prefix-length délka_prefixu
ipv6 nat v6v4 source list natptACL pool název_poolu
```

Máme-li k dispozici pouze jednu veřejnou IPv4 adresu, lze využít NAT s překládáním portů. V tom případě se adresa nedefinuje, určí se přímo IPv4 rozhraní a nakonec příkaz `overload`.

```
ipv6 nat v6v4 source list natptACL interface rozhraní overload
```

Aktuální mapování a statistiky NATu si můžeme prohlédnout pomocí příkazů

```
show ipv6 nat translations
show ipv6 nat statistics
```

Další možnosti nastavení se můžete dočíst ve zdrojích [2], [20], [28] a [29].

# 9 Závěr

V této práci jsem probral obecné vlastnosti a strukturu IPv6 protokolu, který postupně začne nahrazovat IPv4 v celém Internetu. Z popisu je patrné, že IPv6 je vcelku dobře připraveno na provoz izolovaných IPv6 sítí. Také podpora v operačních systémech je dostačující a stále se zlepšuje. Bohužel se soužitím s IPv4 to ještě není tak dobré, existuje sice mnoho přechodových mechanizmů, ale většina má buď nějaké problémy, nebo je komunikace prostřednictvím nich pomalá a nespolehlivá. To je také jedním z důvodu, proč je například ve Windows upřednostňována komunikace po IPv4, pouze pro služby dostupné jen po IPv6 se použije některý z možných přechodových mechanizmů. V současné době se vyplatí zavádění IPv6 ve větší míře pouze v prostředí, kde je jeho nativní podpora od poskytovatele.

Návrh sítě s implementací IPv6 byl realizován v simulačním programu Cisco Packet Tracer, který byl vyvinut firmou Cisco na podporu výuky v kurzech Cisco Networking Academy. Bohužel program Packet Tracer ještě nepodporuje všechny nové funkce a protokoly spojené s provozováním IPv6. Jako simulační nástroj je možné využít také software GNS, který dokáže přímo virtualizovat IOS některých Cisco směrovačů. K tomu je ale třeba mít k dispozici binární soubory IOS, které spadají pod placenou licenci. Podpora IPv6 je pouze v IOS té nejvyšší řady a také ještě ne vždy úplná. Metody, které nebylo možné vyzkoušet v simulačním programu, byly otestovány na reálných zařízeních od firmy Cisco.

## Literatura


[1] DEERING, S.; HINDEN, R. *RFC 2460 - Internet Protocol, Version 6 (IPv6) Specification* [online]. December 1998 [cit. 2011-08-02]. Dostupné z WWW: <http://tools.ietf.org/html/rfc2460>.

[2] SATRAPA, Pavel. *Internetový protokol IPv6* [online]. Praha : CZ.NIC, 2008 [cit. 2011-08-03]. Dostupné z WWW: <http://knihy.nic.cz/files/nic/edice/pavel_satrapa_ipv6_2008.pdf>. ISBN 978-80-904248-0-7.

[3] DUNMORE, Martin (Ed.). *An IPv6 Deployment Guide* [online]. The 6NET Consorcium, 2005 [cit. 2011-08-04]. Dostupné z WWW: <http://www.6net.org/book/deployment-guide.pdf>.

[4] PODERMAŃSKI, Tomáš; GRÉGR, Matěj. IPv6 Mýty a skutečnost, díl V. - Zjednodušené hlavičky. *Lupa.cz* [online]. 10. 3. 2011, [cit. 2011-08-04]. Dostupný z WWW: <http://www.lupa.cz/clanky/ipv6-myty-a-skutecnost-dil-v-zjednodusene-hlavicky/>. ISSN 1213-0702.

[5] PARTRIDGE, C.; JACKSON, A. *RFC 2711 - IPv6 Router Alert Option* [online]. October 1999 [cit. 2011-08-04]. Dostupné z WWW: <http://tools.ietf.org/html/rfc2711>.

[6] JOHNSON, D.; ARKKO, J. *RFC 6275 - Mobility Support in IPv6* [online]. July 2011 [cit. 2011-08-6]. Dostupné z WWW: <http://tools.ietf.org/html/rfc6275>.

[7] HINDEN, R.; DEERING, S. *RFC 4291 - IP Version 6 Addressing Architecture* [online]. February 2006 [cit. 2011-08-5]. Dostupné z WWW: <http://tools.ietf.org/html/rfc4291>.

[8] HINDEN, R.; DEERING, S.; NORDMARK E. *RFC 3587 - IPv6 Global Unicast Address Format* [online]. August 2003 [cit. 2011-08-6]. Dostupné z WWW: <http://tools.ietf.org/html/rfc3587>.

[9] HINDEN, R.; HABERMAN, B. *RFC 4193 - Unique Local IPv6 Unicast Addresses* [online]. October 2005 [cit. 2011-08-6]. Dostupné z WWW: <http://tools.ietf.org/html/rfc4193>.

[10] PODERMAŃSKI, Tomáš. IPv6 Mýty a skutečnost, díl II. - Adresový prostor. *Lupa.cz* [online]. 17. 2. 2011, [cit. 2011-08-05]. Dostupný z WWW: <http://www.lupa.cz/clanky/ipv6-myty-a-skutecnost-dil-ii-adresovy-prostor/>. ISSN 1213-0702.

[11] NARTEN, T.; DRAVES, R.; KRISHNAN, S. *RFC 4941 - Privacy Extensions for Stateless Address Autoconfiguration in IPv6* [online]. September 2007 [cit. 2011-08-07]. Dostupné z WWW: <http://tools.ietf.org/html/rfc4941>.

[12] HUIREMA,C.; CARPENTER, B. *RFC 3879 - Deprecating Site Local Addresses* [online]. September 2004 [cit. 2011-08-07]. Dostupné z WWW: <http://tools.ietf.org/html/rfc3879>.

[13] PODERMAŃSKI, Tomáš; VESELÝ, Vladimír. IPv6 Mýty a skutečnost, díl VII. - Podpora Multicast a anycast provozu. *Lupa.cz* [online]. 24. 3. 2011, [cit. 2011-08-07]. Dostupný z WWW: <http://www.lupa.cz/clanky/ipv6-myty-a-skutecnost-dil-vii-podpora-multicast-a-anycast-provozu/>. ISSN 1213-0702.

[14] NARTEN, T., et al. *RFC 4861 - Neighbor Discovery for IP version 6 (IPv6)* [online]. September 2007 [cit. 2011-08-07]. Dostupné z WWW: <http://tools.ietf.org/html/rfc4861>.

[15] BOUND, J., et al., DROMS, R. (Ed.). *RFC 3315 - Dynamic Host Configuration Protocol for IPv6 (DHCPv6)* [online]. July 2003 [cit. 2011-08-09]. Dostupné z WWW: <http://tools.ietf.org/html/rfc3315>.

[16] PODERMAŃSKI, Tomáš; GRÉGR, Matěj. IPv6 Mýty a skutečnost, díl IV. - Podpora autokonfigurace. *Lupa.cz* [online]. 4. 3. 2011, [cit. 2011-08-08]. Dostupný z WWW: <http://www.lupa.cz/clanky/ipv6-myty-a-skutecnost-dil-iv-podpora-autokonfigurace/>. ISSN 1213-0702.

[17] THOMSON, S.; NARTEN, T.; JINMEI, T. *RFC 4862 - IPv6 Stateless Address Autoconfiguration* [online]. September 2007 [cit. 2011-08-10]. Dostupné z WWW: <http://tools.ietf.org/html/rfc4862>.

[18] DROMS, R. *RFC 3736 - Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6* [online]. April 2004 [cit. 2011-08-10]. Dostupné z WWW: <http://tools.ietf.org/html/rfc3736>.

[19] MALKIN, G.; MINNEAR, R. *RFC 2080 - RIPng for IPv6* [online]. January 1997 [cit. 2011-08-10]. Dostupné z WWW: <http://tools.ietf.org/html/rfc2080>.

[20] BROWN, Sam, et al. *Configuring IPv6 for Cisco IOS*. Rockland : Syngress Publishing, 2002. 362 s. ISBN 1-928994-84-9.

[21] BATES, T., et al. RFC 4760 - Multiprotocol Extensions for BGP-4 [online]. January 2007 [cit. 2011-08-10]. Dostupné z WWW: <http://tools.ietf.org/html/rfc4760>.

[22] ODOM, Wendell. *CCNP ROUTE 642-902 Official Certification Guide*. Indianapolis, USA : Cisco Press, 2010. ISBN: 978-1-58720-253-7.

[23] PODERMAŃSKI, Tomáš; GRÉGR, Matěj. IPv6 Mýty a skutečnost, díl VIII. - Přechodové mechanizmy. *Lupa.cz* [online]. 31. 3. 2011, [cit. 2011-08-09]. Dostupné z WWW: <http://www.lupa.cz/clanky/ipv6-myty-a-skutecnost-dil-viii-prechodove-mechanizmy/>. ISSN 1213-0702.

[24] DAVIS, Joseph. *Understanding IPv6*. Redmond : Microsoft Press, 2008. ISBN: 978-0735624467.

[25] TROAN, O. *Request to move Connection of IPv6 Domains via IPv4 Clouds (6to4) to Historic status* [online]. June 24, 2011 [cit. 2011-08-09]. Dostupné z WWW: <http://tools.ietf.org/html/draft-ietf-v6ops-6to4-to-historic-05>.

[26] SATRAPA, Pavel. Teredo: IPv6 kdekoli. *Lupa.cz* [online]. 18. 11. 2010, [cit. 2011-08-10]. Dostupné z WWW: <http://www.lupa.cz/clanky/teredo-ipv6-kdekoli >. ISSN 1213-0702.

[27] Microsoft. *TechNet* [online]. 2011 [cit. 2011-08-10]. Příkazy Netsh pro rozhraní IPv6. Dostupné z WWW: <http://technet.microsoft.com/cs-cz/library/cc740203(WS.10).aspx>.

[28] EMPSON, Scott; ROTH, Hans. *CCNP ROUTE Portable Command Guide*. Indianapolis : Cisco Press, 2010. ISBN: 978-1-58720-249-0.

[29] *Cisco IOS Configuraion Guide*. Release 12.4. San Jose : Cisco Systems, 2008. 648 s.

## Seznam příloh