Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačního inženýrství

Bakalářská práce

Router s operačním systémem Linux

Lukáš Pospíšil

Zadání

1. Úvod
2. Cíl práce a metodika
3. routování v IPv4 sítích
4. routování v IPv6 sítích
5. Případová studie
6. Závěr
7. Seznam použitých zdrojů

Čestné prohlášení

Prohlašuji, že svou bakalářskou práci "Router s operačním systémem Linux" jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu literatury na konci práce. Jako autor uvedené bakalářské práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 31.3.2011 _____

Poděkování

Rád bych touto cestou poděkoval doc. Ing. Arnoštu Veselému CSc. za poskytnuté cenné rady, které mi velmi usnadnily vypracování této práce.

Obsah

Router s operačním systémem Linux

Souhrn

Tato práce je věnována popisu funkce routeru v síti. Porovnává rozdíly v použití nového protokolu IPv6 vzhledem k staršímu IPv4 protokolu. Tyto rozdíly jsou porovnávány na implementaci IPv6 do již funkčního prostředí.

Klíčová slova: Linux, Router, IPv4, IPv6, Routování, BGP, OSPF, Quagga

---

Router with Unix Operating System

Summary

The thesis focuses on describing how router works. It compares differencies in use of a new IPv6 protocol with an older IPv4 protocol. These differencies are compared on implementation of the IPv6 protocol to a real environment.

Keywords: Linux, Router, IPv4, IPv6, Routing, BGP, OSPF, Quagga

8. Úvod

Tato práce Router s operačním systémem Linux se bude věnovat aktuálnímu problému v prostředí Internetu, přechodu na nový standard IPv6, dále jeho reálnému nasazení a problémům, které toto nasazení přináší a které naopak řeší, to vše s použitím operačního systému GNU/Linux. Tento přechod bude znamenat vyřešení dlouhodobého problému s nedostatkem IP adres, aby Internet mohl dále růst. Pro všechny společnosti připojené do Internetu tím tak vznikl problém, jak se s tím vypořádat.

Tato práce představuje možné problémy při přechodu na nový standard z pohledu malého a z většiny bezdrátového poskytovatele Internetu. Tito malí poskytovatelé zpravidla používají Linux, příp. na Linuxu založený systém pro routování uvnitř a vně své sítě. Linux jakožto vedoucí operační systém na serverech v sobě skrývá schopnosti fungovat jako router nebo také aplikační, databázový či proxy server, jeho využití je široké a jako router je plně funkční. Ovšem i tak jsou zde určité problémy, které při přechodu na nový standard adres, bude nutné vyřešit. Jaké problémy to jsou, se tato práce pokusí zmapovat. Objasní také, které stávající problémy naopak nový standard řeší.

V práci je též zachycen vývoj, který tento stavební protokol Internetu za svou dlouhou historii prodělal a jaké jsou hlavní rozdíly mezi starým IPv4 a novým IPv6 standardem. Nový standard byl vyvíjen s ohledem na nedostatky starého standardu, s ohledem zlepšit situaci v rámci internetu a také se pokusit poskytnout zjednodušení a zrychlení práce routerům. V případové studii pak představí implementaci IPv6 routování za pomoci Linuxu v již fungující síti malého poskytovatele internetu. je zde názorně předvedena snadnost rozchození IPv6 routování a jeho implementace do páteřní sítě. Autor je členem týmu lidí, kteří mají na starosti tuto implementaci, představení adresního plánu pro IPv6 a vypracování vzorových konfigurací pro jednotlivé prvky v síti.

9. Cíle práce a metodika

Tato práce si dává za cíl popsat práci routeru, přiblížit vývoj internetu jako takového. Objasní principy, které se používají pro komunikaci v rámci celého Internetu a představí starý a nový standard. Oba tyto standardy též mezi sebou porovná s poukázáním na možnosti budoucího vývoje. V praktické části pak představí funkční řešení routovaní vnitřní sítě konkrétního poskytovatele internetu a také řešení routování směrem k ostatním subjektům v prostředí Internetu, vše za použití operačního systému GNU/Linux.

V první kapitole je zachycen vývoj internetu, s jakými předpoklady byl spuštěn, jaká je situace nyní. V této kapitole jdou též popsány procesy, které probíhají při komunikaci koncového počítače skrze síť poskytovatele internetu a pak i v rámci internetu směrem k cílové službě *(webový server, herní server, atd.)*. Zmíněny jsou též některé přicházející problémy spojené s nedostatkem IPv4 adres a jejich možná řešení.

Druhá kapitola představí hlavní rozdíly nového IPv6 protokolu oproti starému IPv4 z předchozí kapitoly. Shrne hlavní důvody, které stály u jeho vzniku. Přiblíží změny, které se již udály za jeho krátkou historii. Zmíní též jaké přináší výhody, v čem zjednodušuje práci správcům sítě a naopak, jaké vlastností pozbývá oproti staršímu protokolu. Jelikož jde o protokol relativně nový a ostrá nasazení se v současné době u většiny ISP teprve dokončují, zmíní tato kapitola také další vývoj IPv6 protokolu.

Případová studie s využitím poznatků z předchozích kapitol obsahuje funkční návrh řešení routování páteřní sítě jednoho konkrétního poskytovatele internetu. Tento návrh bude obsahovat představení již fungujícího IPv4 routování a následně implementaci routování IPv6 do tohoto prostředí. V této kapitole též autor shrne osobní zkušenosti s novým protokolem, který získal při jeho zavádění do provozu.

V závěru dojde k shrnutí celkových pro a proti nového protokolu. Budou zodpovězeny otázky, které si autor položil v úvodní kapitole. A bude zde zhodnoceno celkové splnění všech požadavků.

10. Routovaní v IPv4 sítích

10.1 Historie Internetu

Počátky internetu sahají do 60. let 20. století, kdy se americká armáda snažila vyřešit problém s propojením svých počítačů tak, aby v případě výpadku jednoho či více uzlů, například z důvodu nukleárního útoku, síť i nadále fungovala. Bylo tedy důležité, aby síť jako taková byla decentralizovaná a komunikace probíhala na základě aktuálního stavu sítě. Přišlo se tedy s myšlenkou paketově orientované komunikace, kdy by každý jednotlivý datový paket obsahoval dostatek informací, aby se sám dostal bez problému k cíli. Tento způsob komunikace, ač se zdá být neefektivní z pohledu nedržení stálého spojení mezi komunikujícími uzly, jak je tomu například u telefonní sítě, zajišťuje síti nebývalou robustnost. V té době se tato síť propojující zatím pouze výkonné počítače amerických univerzit jmenovala ARPANET. Původně tato síť sloužila ke sdílení procesorového času připojených superpočítačů, až časem se začala využívat jako informační médium – emaily a mailling listy[1].

Jak se síť postupně rozrůstala bylo nutné nahradit původní komunikační protokol jiným a lepším, a tak byl vynalezen komunikační standard TCP/IP, nejde o jediný protokol, nýbrž o rodinu více než 100 protokolů, která zabezpečuje a standardizuje komunikaci v rámci Internetu. Jde o standard otevřený a každý výrobce operačního systému tak mohl podporu TCP/IP začlenit, a proto se stala síť ještě více decentralizovanou či spíše postupně čím dál více neřízenou ve svém rozvoji. Kdokoli se tak mohl na své náklady připojit a nebyla zde žádná vyšší autorita, která tomu mohla zabránit. Toto zapříčinilo odchod armádních organizací začátkem 80. let z této, stále ARPANET pojmenované, sítě. V průběhu 80. let, jak se počet přípojených sítí zvětšoval, se začal pro tuto síť používat název Internet. Začaly postupně přibývat další služby, jaké známe dnes, například world wide web počátkem 90. let, a Internet se tak stal primárním médiem pro komunikací na světě[2].

10.2 Jak pracuje router?

Abychom pochopili funkci Linuxu na routeru, je třeba si nejdříve vysvětlit, jak takový router pracuje. Router má zpravidla na starosti směrování paketů z jedné sítě do

1 BARTOŠEK MIROSLAV, Krátce z historie Internetu, [online]

2 BARTOŠEK MIROSLAV, Krátce z historie Internetu, [online]

druhé či obecněji od zdroje k cíli. Pokud se na funkci routeru podíváme z hlediska OSI modelu a síťových protokolů jedná se o L3 *(Layer 3 - třetí vrstvu)* a IP protokol[3].

10.2.1 OSI model síťové komunikace

OSI model se používá pro abstraktní popis síťové komunikace, kterou rozděluje do sedmi vrstev. Při přechodu paketu z jedné vrstvy na druhou se používá enkapsulace *(zapouzdření)*. To znamená, že datový paket, konkrétně data která obsahuje, jsou obalena hlavičkami konkrétních síťových vrstev a prvky, ať už jde o routery, switche, na jednotlivých síťových vrstvách pracují pouze s tím, co jim přísluší a při průchodu paketu skrze sebe paket rozbalí, příslušnou hlavičku přečtou a případně pozmění, zabalí a pošlou na další uzel směrem k cíli. Toto přebalování v rámci Internetu, konkrétně protokolů TCP/IP, nejčastěji zastávají routery neboli směrovače[4].

TCP / IP model	OSI model	Příklad
Aplikační	Aplikační	SSL, HTTP, DNS
	Prezentační	
	Relační	
Transportní	Transportní	TCP, UDP
Internet	Síťová	IP, ICMP, OSPF
Linková a fyzická	Linková	Ethernet, ARP
	Fyzická	

Tabulka 1: OSI model síťové komunikace

Tato abstrakce se ovšem v praxi moc nevyužívá, je dobrá pro pochopení obecných principů, ale při použití nějakého protokolu z rodiny TCP/IP je výhodnější se držet praktičtějšího rozdělení do čtyř vrstev. S tím, že tento zjednodušený model obsahuje pouze čtyři vrstvy: Aplikační, Transportní, Síťovou a vrstvu síťového rozhraní[5].

10.2.2 IP protokol

Pokud se podíváme konkrétně na IP protokol, tak ten v celé problematice zaujímá nejdůležitější roli, protože na jeho základě mezi sebou komunikují jednotlivé sítě mezi

3 BOUŠKA PETR, Víte jak pracuje router?, [online]

4 DOSTÁLEK LIBOR, KABELOVÁ ALENA, Velký průvodce protokoly TCP/IP a systémem DNS, s. 6

5 BOUŠKA PETR, Víte jak pracuje router?, [online]

sebou. IP protokol je zkratka pro Internet Protokol a jeho zásluhou dostal Internet své jméno.

Jde o nespojitý protokol – při putování paketů mezi odesílatelem a příjemcem se nevytváří ustálené spojení a každý paket obsahuje veškeré informace potřebné k tomu, aby dosáhl adresy příjemce. Také se může stát, že ne každý paket půjde stejnou cestou, tudíž je více než možné, že dorazí v jiném pořadí, než jak byl odeslán. Je to řešeno číslováním posloupnosti jednotlivých paketů dané komunikace.

Celý IP protokol je tedy tvořen dílčími protokoly[6]:

- vlastním protokolem IP (IPv4 a IPv6)
- služebními protokoly ARP a RARP
- servisními protokoly ICMP a IGMP

10.2.3 IPv4 adresy

Protože v linkové vrstvě *(Layer 2 – druhá vstva)* má každé síťové rozhraní fyzickou adresu *(MAC adresa)*, tak i u IP protokolu bylo využito podobného principu. V případě IPv4 je čtyřbajtová, nejčastěji zapisovaná v desítkové soustavě s bajty oddělenými tečkami, přičemž využitelných adres je 4.3×10^9 tj. 4,3 Mld. IPv4 adresy jsou až na pár výjimek unikátní pro jednotlivá síťová rozhraní v rámci celého Internetu. Následující tabulky obsahují popis nejdůležitějších rozsahů IPv4 adres a dále je jejich význam blíže popsán[7]:

Privátní IPv4 adresy			
Síť	**Adresa sítě**	**Broadcast adresa**	**Adresní rozsah**
10.0.0.0/8	10.0.0.0	10.255.255.255	10.0.0.1 – 10.255.255.254
172.16.0.0/12	172.16.0.0	172.31.255.255	172.16.0.1 – 172.31.255.254
192.168.0.0/16	192.168.0.0	192.168.255.255	192.168.0.1 – 192.168.255.254

Tabulka 2: Přehled privátních IPv4 adres[8]

6 DOSTÁLEK LIBOR, KABELOVÁ ALENA, Velký průvodce protokoly TCP/IP a systémem DNS, s. 119

7 DOSTÁLEK LIBOR, KABELOVÁ ALENA, Velký průvodce protokoly TCP/IP a systémem DNS, s. 119

8 BOUŠKA PETR, Adresování v IP sítích, 21.7.2010, [online]

Veřejné IPv4 adresy				
Třída	Rozsah IP	Maska sítě	CIDR maska	Poznámka
A	0.0.0.0 – 127.255.255.255	255.0.0.0	/8	Hlavní
B	128.0.0.0 – 191.255.255.255	255.255.0.0	/16	Hlavní
C	192.0.0.0 – 223.255.255.255	255.255.255.0	/24	Hlavní
D	224.0.0.0 – 239.255.255.255			Multicast
E	240.0.0.0 – 255.255.255.255			Rezervováno

Tabulka 3: Rozdělení IPv4 adres podle tříd[9]

Rozdíl privátních a veřejných IPv4 je ten, že veřejné jsou propagované v rámci celého internetu – jednoduše: počítače s veřejnou IPv4 adresou lze kontaktovat v rámci celého internetu, naopak privátní adresy jsou nedosažitelné a nesmějí se vůbec objevit v globálních routovacích tabulkách. Proto lze privátní adresy používat na více místech najednou, například za pomoci NAT překladu adres, kdy router má veřejnou adresu a zbytek vnitřní sítě používá privátní adresy. Privátní adresy tak mohou a jsou jen mizivým procentem v rámci celkového IPv4 adresního prostoru, přesto tyto adresy skutečně používá momentálně obrovské množství počítačů a tak ne každý počítač je z tohoto důvodu skutečně členem internetu. Tento problém faktické nedosažitelnosti, který byl ostatně jednou ze základních myšlenek vzniku internetu a nyní je čím dál více porušován, se snaží řešit nový IP protokol – IPv6[10].

V minulosti byly adresy přidělovány podle tříd (A až C) s pevnou délkou masky, ale i když se přidělovalo podle velikosti žadatele, tak byly bloky značně rozdílných velikostí *(třída A – 16 mil., třída B – 65 tis., třída C – 256 adres)*, to se později ukázalo jako neefektivní a značně plýtvající IP adresami. Tak se přešlo na systém s proměnou délkou masky CIDR. V současné době, kdy adresy docházejí, dostane žadatel počet IPv4 adres, které mu vystačí na 8 měsíců a tento počet měsíců se bude dále snižovat, takto zní poslední vydaná pravidla RIPE k přidělování IPv4 adres. Například jedno z

9 BOUŠKA PETR, Adresování v IP sítích, 21.7.2010, [online]
10 BOUŠKA PETR, Adresování v IP sítích, 21.7.2010, [online]

posledních větších přidělení v rámci členů českého internetu – NIXu je prefix o velikosti /20 *(4096 IPv4 adres)*. Žadatel po jejich vyčerpání může žádat o další přidělení, ovšem z důvodu vyčerpání IPv4 jde nejspíše o poslední přidělení pro daného žadatele vůbec.

Třídy A až C jsou tedy jasné, třída D je multicast, což je speciální typ adres používaný ke speciálnímu způsobu doručování dat více příjemcům najednou, tak, že se odešlou data z jednoho zdroje a pakety se distribuují mezi více příjemců. Multicastového síření paketů využívá se například u IPTV, více viz. kapitola 3.2.7 IGMP[11].

Mezi další typy, kromě již zmíněného multicastu, patří:

- **Unicast** – tento způsob označuje zasílání paketů jedinému cíli, jde o nejčastější případ komunikace
- **Broadcast** – zasílání paketů všem v daném subnetu, tomuto způsobu komunikace je vždy vyhrazena poslední IP adresa z daného subnetu, ale z důvodu častého zneužití k síťovým útokům
- **Loopback –** tento typ adres z rozsahu 127.0.0.0/8 se používá u počítačů ke komunikaci sám se sebou[12].

10.2.4 Komunikace v IP sítích

V IP sítích se používá jako stavebního prvku zařízení zvané router, což v řadě případů není nic jiného než počítač s více sítovými rozhraními a nějakou Linuxovou distribucí. Pro profesionální a náročnější úlohy se používají speciálně vyráběné boxy, které jsou daleko lépe připravené na routování paketů, jelikož routování zabezpečují pomocí drahých hardwarových přídavných karet. Oproti softwarovému řešení dosahují mnohem větších rychlostí a kratší doby zpracování. Routery mají za úkol mezi sebou propojit jednotlivé LAN *(Local Area Network)*[13] sítě do WAN *(Wide Area Network)*[14] sítí, kdy nejznámější WAN síť není nic jiného než Internet.

Celá komunikace probíhá následovně - odesílatel *(osobní počítač)* se potřebuje spojit s příjemcem v jiné síti *(webový server)*, pošle datový paket po linkové vrstvě

11 BOUŠKA PETR, Adresování v IP sítích, 21.7.2010, [online]
12 BOUŠKA PETR, Adresování v IP sítích, 21.7.2010, [online]
13 LAN *(Local Area Network)*, označuje síť, která pokrývá malé úzení. Například jednu budovu společnosti, jednu domácnost a její území končí za prvním routerem.
14 WAN *(Wide Area Network)* je, jak je již z názvu patrné, síť která pokrývá velké území. Nejznámější WAN sítí je Internet, z českých neveřejných WAN sítí například CZFree.Net.

routeru *(směrovači)* obsluhující danou LAN. Router paket vybalí z linkového rámce. Poté se router podívá do IP-datagramu na cílovou IP adresu paketu, záznam porovná s vlastní routovací tabulkou a podle toho vybere jakým síťovým rozhraním vypustí paket – tj. do jakého linkového rámce opět paket zabalí. Při přebalování ještě zmenší hodnotu TTL *(Time To Live)* v hlavičce datagramu o minimálně 1, jedná se o bezpečnostní mechanismus, aby Internetem nekolovaly zacyklené pakety – jakmile TTL dosáhne hodnoty 0, paket se zahodí. Takto paket putuje mezi jednotlivými routery, stále se přebaluje do jiného linkového rámce, TTL se snižuje, a to vše dokud paket nedorazí ke svému cíli.

Konkrétní způsoby a náležitosti této komunikace pro každý jednotlivý TCP/IP protokol jsou popsány v RFC[15] dokumentech. Každý výrobce operačního systému tak má přístup ke specifikaci komunikace a může ji začlenit tak, aby všechny operační systémy byly v této komunikaci vzájemně kompatibilní, což se zatím výrobcům daří, i když někteří ne vždy a kompletně daná RFC respektují.

10.2.5 ARP

Protokol ARP *(Address Resolution Protocol)* stojí trochu stranou od zbylých IP protokolů, protože nevyužívá IP hlavičku, ale je balen pouze do linkového rámce. Jeho funkce, ač nemá moc společného s routováním je důležitá pro pochopení dalších procesů. Využívá se při komunikaci mezi jednotlivými prvky v LAN, kdy odesílatel zná svou linkovou *(fyzickou)* adresu, svou IP adresu a IP adresu cíle. Nezná ovšem fyzickou adresu cíle, tak využije protokol ARP pro její zjištění. Zašle dotaz na všeobecnou linkovou adresu *(FF:FF:FF:FF:FF:FF)* a vrátí se mu odpověď, výsledek si uloží do své ARP tabulky *(ARP cache)*, což je jednoduchá tabulka, kde je každé fyzické adrese *(MAC)* přiřazena odpovídající IP adresa, případně hostname a síťové rozhraní, kde se daný prvek nachází. Pokud se cílový prvek nachází mimo LAN, odesílající prvek použije linkovou adresu routeru, obsluhující danou LAN[16].

server.lucky.jiznak.czf (10.38.21.1) at 0:22:15:b9:91:aa on en0 ifscope [ethernet]
router.lucky.jiznak.czf (10.38.21.6) at 0:c:42:16:b5:46 on en0 ifscope [ethernet]
lucky15.lucky.jiznak.czf (10.38.21.15) at ff:ff:ff:ff:ff:ff on en0 ifscope [ethernet]

15 RFC *(Request for comments)* označuje typ dokumentů, které popisují řadu internetových protokolů, standardů, postupů, návodů, ovšem mají spíše formu doporučení než normy, ale aktuální situace je většinou taková, že je pravidlem je dodržovat a porušení RFC je výjimkou.

16 DOSTÁLEK LIBOR, KABELOVÁ ALENA, Velký průvodce protokoly TCP/IP a systémem DNS, s. 119

Z bezpečnostního hlediska toto řešení není zcela optimální, protože se může útočník pokusit o podvrhnutí ARP odpovědí a prohlásit se routerem nebo vystupovat za jiný počítač přítomný na LAN a přijímat pak za něj data. Toto lze, i když ne úplně řešit filtrováním ARP tabulky, kdy nastavíme záznamy statické, takže do nich nelze přidávat další (podvrhnuté, ale i pravdivé) ARP záznamy.

RARP

Tento protokol používá naopak obrácený postup – RARP *(Reverse Address Resolution Protocol)*, používá se zpravidla u bezdiskových stanic, které znají svou linkovou adresu, ale neznají k tomu svou příslušnou IP adresu. Pošlou tedy dotaz – mám linkovou adresu xy, jaká je moje IP adresa a přítomný RARP server IP adresu přidělí a zašle odpověď. Tento protokol se v současné době téměř nevyužívá, protože existuje komplexnější náhrada – DHCP[17].

10.2.1 ICMP

Pokud ovšem dojde při cestě paketu k cíli k nějaké mimořádné situaci, využije router tento servisní protokol. Protokolem ICMP lze oznamovat nejrůznější chyby v síťové komunikaci, ovšem nezřídka se stává, že různé operační systémy či síťové prvky podporují jen část těchto signálů a navíc některé routery z bezpečnostních důvodů určité typy ICMP paketů rovnou blokují. Proto si popíšeme jen ty nejčastěji používané[18]:

Echo

Ne vždy se pomocí ICMP oznamují chyby, a tuto výjimku představuje právě typ Echo. S jeho pomocí se zjišťuje dosažitelnost nějakého cílového uzlu v Internetu. Žadatel odešle ICMP paket „žádost o echo“ a příjemce (pokud je dosažen) mu odpoví „echo“. V Unixových systémech k tomuto slouží příkaz ping, který navíc počítá za jak dlouho od cílového uzlu přijde paket s odpovědi „echo“, případně poskytuje další volby.

Nedoručitelný IP-datagram

Tento typ oznamuje chybu o nedoručitelnosti IP-datagramu. Pokud nějaký uzel po

17 DOSTÁLEK LIBOR, KABELOVÁ ALENA, Velký průvodce protokoly TCP/IP a systémem DNS, s. 119

18 DOSTÁLEK LIBOR, KABELOVÁ ALENA, Velký průvodce protokoly TCP/IP a systémem DNS, s. 119

cestě směrem k příjemci zjistí, že daný paket nelze doručit, pošle o této situaci odesílateli oznámení. Důvodů muže být více, např. nedosažitelný uzel, síť, port, chybné směrování, a jiné.

Sniž rychlost odesílání

Pokud je síť po cestě k cíli v nějakém místě přetížena a daný směrovač nestíhá posílat pakety dále k síti, zašle zprávu „sniž rychlost odesílání“, pokud odesílatel využívá služeb TCP protokolu, tak reaguje snížením rychlosti odesílání, ovšem v případě UDP protokolu jsou tyto oznámení ignorovány[19].

Čas vypršel

Jak bylo zmíněno výše, pokud se u IP-datagramu sníží položka TTL na nulu, dojde k jeho zahození. Takové zahození je pak signalizováno typem s kódem nula. Typem čas vypršel se dále signalizuje zcela jiná situace – adresát není schopen v daném čase sestavit z fragmentů celý IP-datagram (kód = 1). Typ s kódem = 0 využívá v Unixu program traceroute, který slouží k odhalení informací o routerech na cestě mezi odesílatelem a příjemcem[20].

Fragmentace paketů

Fragmentace paketů, se využívá pokud daná linková vrstva nemá dostatečnou kapacitu *(MTU)* pro průchod celého paketu, tak ten se rozdělí na více paketů, které se pak odděleně zasílají dál k cíli a ten si je pak sestaví v kompletní IP-datagram. Jednotlivé fragmenty jsou ještě označeny pořadím, protože IP-protokol negarantuje, že půjdou stejnou cestou a tudíž, že dorazí ve stejném pořadí. Toto platí pro pakety označené příznakem - fragmentace možná, je tu ale i možnost fragmentaci zakázat a paket nezbývá než zahodit, což je pak odesílateli signalizováno příslušnou ICMP zprávou[21].

MTU

MTU *(Maximum Transfer Unit)* znamená maximální velikost paketu v Byte, který je možné přes danou linkovou vrstvu poslat. Ethernet zpravidla má MTU o velikosti

19 DOSTÁLEK LIBOR, KABELOVÁ ALENA, Velký průvodce protokoly TCP/IP a systémem DNS, s. 119

20 DOSTÁLEK LIBOR, KABELOVÁ ALENA, Velký průvodce protokoly TCP/IP a systémem DNS, s. 119

21 DOSTÁLEK LIBOR, KABELOVÁ ALENA, Velký průvodce protokoly TCP/IP a systémem DNS, s. 119

1500 B, více v následující tabulce.

Médium	MTU (Byte)	Poznámka
Ethernet	1500	Standardní velikost MTU
WLAN (802.11)	2272	
IPv4	Minimálně 576	Každý uzel musí být schopen přijmout paket s min. velikostí 576 B
IPv6	Minimálně 1280	Pro zjištění MTU lze využít funkce objevování MTU cesty

Tabulka 4: Přehled MTU pro určitá přenosová média

10.2.1 IGMP

Protokol IGMP má na starosti posílání adresních oběžníků *(mutlicasts)* v rámci LAN. U IGMP paketů je TTL nastaveno na hodnotu 1, tak aby bylo zajištěno, že jsou pouze pro danou LAN. V rámci internetu jsou některé směrovače součástí tzv. multicast backbone, která zabezpečuje šíření adresních oběžníků.

Pomocí protokolu IGMP se pak posléze zajišťuje jejich dopravení konkrétním stanicím v rámci LAN, řeší se jím vše od registrace do skupiny – stanice je připravena pro šíření konkrétního adresního oběžníku. Přes zjišťování, zda stanice má stále zájem tyto pakety přijímat, což je důležité z důvodu toho, aby se zbytečně LAN nezahlcovala daty, které již nikdo nepotřebuje. Až po zastavení přijímání dalším typem IGMP paketu, kdy stanice odešle požadavek o ukončení přijímání[22].

10.3 Způsoby routování IP paketů

IP protokol máme tedy již popsaný, stejně tak způsob, jakým pracuje router s jednotlivými pakety. Existuje ovšem více způsobů jak na dané routování pohlížet, především z pohledu správy routovací tabulky, a tomu se bude věnovat následující kapitola. Základní rozdělení je na statické a dynamické routovaní.

10.3.1 Statické routovaní

Statické routovaní znamená, že síťový administrátor na každém routeru ručně vyplní routovací tabulku, podle které se router řídí při směrování paketů k cíli. V

22 DOSTÁLEK LIBOR, KABELOVÁ ALENA, Velký průvodce protokoly TCP/IP a systémem DNS, s. 119

případě malé sítě a malém počtu routerů, které ji obsluhují, není problém ručně udržovat jednotlivé routovací tabulky. V případě sítě větší to již problém je a v té chvíli je více než nutné využít služeb některého protokolu zabezpečující dynamické routovaní. Toto řešení je bezpečné a do určité míry přehledné, ale již ze své podstaty nereflektuje změny v síti.

10.3.2 Defaultní routování

Jde o speciální případ routování, kdy router nemá ve své tabulce odpovídající záznam pro cíl daného paketu, tak použije defaultní cestu, která slouží jako taková poslední záchrana, aby se nestávalo, že router se nemůže rozhodnout kam má daný paket poslat[23].

10.3.3 Dynamické routovaní

V případě, že máme síť většího rozsahu, stálo by statické udržování routovacích tabulek routerů mnoho času, proto se ve větších sítích využívá služeb dynamického routování. Jeho hlavní výhody jsou zřejmé[24]:

- reaguje na změny v síti a přizpůsobuje jim routovací tabulku
- automaticky vypočítává optimální cesty skrze síť
- zvyšuje robustnost sítě a její odolnost vůči výpadkům

Má ale také určité nevýhody:

- méně bezpečné, nutnost věnovat čas zabezpečení a příp. ověřování routerů v síti
- z počátku složitější na konfiguraci
- náročnější na výpočetní výkon routeru

Dynamické routovací protokoly dělíme podle použití na:

- interior gateway protocols – IGP – routuje se jimi uvnitř sítě (AS)
- exterior gateway protocols – EGP – routuje se s nimi mezi jednotlivými sítěmi (AS)[25]

23 BOUŠKA PETR, Víte jak pracuje router?, [online]

24 PETR EMANUEL, Implementace IPv6, [online]

25 AS *(autonomous system)* je označení pro síť, která má jednotnou správu a pravidla a spravuje ji konkrétní autorita. V

IGP protokoly

Jak již bylo řečeno, tyto protokoly se používají pro routování uvnitř AS, mezi tyto protokoly patří: OSPF, IS-IS, RIP/v2 a (E)IGRP.

OSPF patří mezi nejrozšířenější IGP protokoly. Jde o otevřený protokol, který podporuje dnešní classless *(CIDR)* systém podsítí. Pro výpočet nejkratší cesty používá Dijkstrův algoritmus SPF – Shortest Path First a je založen na stavu linek. To znamená, že každý router si udržuje aktuální mapu sítě a v případě změny si routery navzájem pošlou zprávu o jakou změnu jde. Router z této mapy, která kromě informací, kdo je s kým propojen, obsahuje také cenu linek, prefixy jednotlivých podsítí a jiné údaje, vypočítá strom nejkratších vzdáleností ke všem známým cílům, kterých je on sám kořenem. Tím zjistí, kudy vedou nejkratší cesty ke všem cílům a ty pak zanese do své routovací tabulky. Hlavní výhodou OSPF je rychlá reakce na změny v síti a díky možnosti síť virtuálně rozdělit na logické celky – arey, zajistí routování i v rozsáhlých sítích. V rámci takové arey se udržuje mapa sítě pouze pro ni a v případě, že nejde o backbone areu (0.0.0.0) ještě cesta k nejbližšímu routeru z backbone arey. Pro šíření údajů mezi routery se používá multicast a metoda zaplavování. K šíření takových údajů je nejdříve nutné, aby se routery staly sousedy, což je zabezpečeno pomocí pravidelných hello paketů *(zpravidla 10s)*. V případě shodných údajů *(číslo oblasti, typ oblasti, subnet s maskou, atd.)* naváže router sousedství s druhým a můžou si vyměňovat vzájemně informace o síti. OSPF ve verzi 2 se používá pro IPv4 routování a v novější verzi 3 již přibyla podpora IPv6[26].

IS-IS *(Intermediate system to intermediate system)* je protokol velice podobný OSPF, používá též Dijkstrův algoritmus pro výpočet cesty v síti a je také založený na stavu linek, dokonce reaguje ještě rychleji na změny v síti než OSPF. Byl vyvinut firmou DEC a později ho převzala ISO organizace za svůj routovací protokol k referenčnímu OSI modelu. S jeho neúspěchem ovšem následoval i neúspěch IS-IS a komunita přišla s podobným protokolem – OSPF. Ač se zdají být v podstatě stejné, významný rozdíl mezi nimi je u vnímání jednotlivých oblastí *(areas)*, kdy OSPF dělí oblasti již na jednotlivých síťových rozhraních routeru, tak u IS-IS hranice oblastí procházejí linkami, tedy každý router patří do určité oblasti jako celek. Tento protokol

rámci Internetu jsou AS stavebním prvkem a podléhají registraci, v rámci českého NIXu jsou jimi například UPC.cz, Cesnet, aj.

26 SATRAPA PAVEL, IPv6, Internetový protokol IPv6, s. 136

podporuje též IPv6[27].

RIP/v2 *(Routing Information Protocol)* je již celkem starý routovací protokol, opět jde o otevřený standard a vyniká svou jednoduchostí, nemá hello pakety. Bohužel tato jednoduchost přináší i řadu nevýhod a tou je především pomalá reakce na změny v síti a omezení ze strany maximálního počtu hopů na 15. Cena cesty je založena na vektoru vzdáleností, kdy linky mají stanovenou určitou cenu *(nejčastěji 1)* a ta se postupně sčítá, do hodnoty 15 je vše v pořádku, ale hodnota 16 již znamená nedosažitelný cíl. Tento protokol je tedy vhodný spíše pro malé sítě. Ve verzi RIPng podporuje IPv6[28].

(E)IGRP (*Enhanced Interior Gateway Routing Protocol)* je proprietární cisco protokol , který ve své rozšířené verzi podporuje vlastnosti protokolů založených jak na vektoru vzdáleností, tak na stavu linek. Na změny v síti reaguje svižně a pro výměnu informací mezi routery používá též multicast, v rozšířené verzi podporuje též IPv6[29].

EGP protokoly

Tuto skupinu zastupuje v dnešní době jediný protokol a to BGP ve verzi označované jako BGP4+, v této verzi je již zahrnuta podpora pro IPv6 protokol. BGP tedy vděčíme za to, že se připojíme na náš oblíbený webový server či si vybereme emailovou schránku, routuje se s ním v rámci celého Internetu. Jelikož se s ním routuje mezi jednotlivými AS, jde o protokol, který spoléhá daleko více na práci administrátora než například OSPF, třeba sousedství je nutné ručně nastavit. Mezi jednotlivými sousedy si udržuje stabilní TCP spojení a při té příležitosti si vymění kompletní routovací údaje, a to i včetně získaných od sousedů. Poté si již posílají pouze aktualizace. Takto si to předávají další BGP routery dál mezi sebou a každý BGP router tak zjistí, co je kudy dostupné. Problém nastává v případě, že se TCP spojení rozpojí, oba BGP routery prohlásí svého souseda za nedosažitelného a odstraní z routovacích tabulek všechny údaje jím předané a tím též dojde k přepočtu routovacích tabulek mezi routery. Tento výpočet je vcelku náročný a i takový hardwarový router Cisco c7604 dokáže s plnými BGP tabulkami zaměstnat i na několik minut, proto v rámci českého Internetu funguje mailing-list, na který členové NIXu oznamují předem technické zásahy a případnou nedostupnost svých BGP routerů[30].

27 SATRAPA PAVEL, IPv6, Internetový protokol IPv6, s. 144
28 SATRAPA PAVEL, IPv6, Internetový protokol IPv6, s. 130
29 BOUŠKA PETR, Cisco Routing 6 – srovnání routovacích protokolů, [online]
30 SATRAPA PAVEL, IPv6, Internetový protokol IPv6, s. 147

Protokol BGP na rozdíl od IGP protokolů nepoužívá pro výpočet cesty běžné typy metrik, ale má složitější systém rozhodování podle různých kritérií. Rozhoduje se podle cesty, politik a pravidel, z toho pak vybere nejlepší cestu k cíli[31].

BGP router s routovacími informacemi pracuje pomocí tří bází:

- **Vstupní báze** – obsahuje informace, které router obdržel od některého ze svých sousedů, ty jsou posléze na základě pravidel posouzeny a poté je na jejich základě modifikována lokální báze
- **Lokální báze** – představuje routovací tabulku BGP routeru podle níž se rozhoduje kam se daný paket pošle
- **Výstupní báze** – zahrnuje informace, které router ohlašuje svým sousedům

BGP ale umožňuje daleko více. Řada správců například nastavuje pro sousedy filtry, které zamezí předání nesprávného prefixu dál do Internetu či lze použít tzv. komunity, což naopak slučuje prefixy podle určité podobnosti, například prefixy českého Internetu – NIX, a podle toho pak dané cesty upřednostňovat před zahraničními prefixy. Pokud BGP router přebírá plnou tabulku prefixů, jejich číslo se u IPv4 blíží ke 300 000. U IPv6 je to přibližně 20 000 prefixů. BGP router lze také nakonfigurovat jako route-reflektor, což se využívá třeba v českém NIXu, ale i dalších národních uzlech. Route-reflektor je BGP router, který pouze předává naučené routy, výhoda je pak v tom, že nemusí každý s každým udržovat sousedství, ale stačí mít sousedství s route-reflektorem[32].

31 BOUŠKA PETR, Cisco Routing 6 – BGP – Border gateway protocol, [online]
32 SATRAPA PAVEL, IPv6, Internetový protokol IPv6, s. 147

11. Routování v IPv6 sítích

V předchozí kapitole byly popsány principy routování v IPv4 sítích. Jelikož nový protokol spíše řadu věcí upravuje, než mění, bude tato kapitola věnována rozdílům IPv6 oproti dosavadnímu IPv4 protokolu.

11.1 Základní vlastnosti IPv6

- obrovský adresní prostor – přibližně $6{,}67\times10^{23}$ adres na čtvereční metr povrchu Země
- ICMPv6 sjednocuje funkce a schopnosti ARP a ICMP
- bezstavová autokonfigurace adres
- vestavěné bezpečnostní mechanismy (IPsec)
- optimalizace routování – fixní velikost základní hlavičky 40B, další rozšiřující hlavičky, směrovače neprovádí fragmentaci
- MTU na lince musí být minimálně 1280B, detekci MTU dělá odesílatel, stejně tak i fragmentaci paketů
- broadcast adresy zrušeny, nahrazeny multicastovou skupinou all-hosts[33]

11.2 Typy a rozdělení IPv6 adres

IPv6 používá podobných principů adresování jako IPv4, ale jelikož má primární úkol řešit nedostatek IP adres, tak oproti IPv4 je její adresa šestnáctibajtová (přibližně adres: 3.4×10^{38}) – tj. prefix /128 značí jedinou adresu a minimální prefix přidělitelný na síťové rozhraní o velikosti /64 je 2^{64} adres. Je to přímý nástupce IPv4 protokolu a v současné době se hromadně nasazuje, jelikož volných IPv4 adres už moc nezbývá a pomalu ale jistě začíná být nutné provozovat služby i na IPv6. Jeho základní nevýhoda je ovšem v nekompatibilitě s IPv4, což v praxi znamená, že sítě jsou zpravidla dual-stackové, tzn. na síti probíhá oddělená komunikace pro IPv4 a IPv6[34].

33 PETR EMANUEL, Implementace IPv6, 2011, [online]

34 SATRAPA PAVEL, IPv6, Internetový protokol IPv6, s. 147

Následující tabulka obsahuje přehled jednotlivých typů IPv6 adres:

Prefix	**Význam**
::/128	Nedefinovaná adresa
::1/128	Lokální smyčka (loopback)
fc00::/7	Unikátní individuální lokální
fe80::/10	Individuální lokální linkové
ff00::/8	Skupinové adresy (multicast)
ostatní	Individuální globální (unicast)
Prefix:0:0:0:0	Výběrové adresy (anycast)

Tabulka 5: Typy IPv6 adres

Lokální smyčka *(loopback)* slouží obdobně jako u IPv4 k vnitřní komunikaci v rámci počítače. Rozdíl oproti IPv4 je především v absenci broadcast adres, což má hlavně bezpečnostní důvod – u IPv4 sítí je jeden z možných útoků tzv. broadcast storm, kdy se útočník snaží o zahlcení LAN pomocí chybných dotazů na broadcast adresu dané podsítě *(subnetu)*. Na dotaz na tuto adresu by měly odpovědět všechny aktivní stanice dané podsítě, pokud ovšem odpoví zase broadcastem, vznikne chaos a síť se pomalu zahltí. Takže jeho potřeby převzaly adresy skupinové, což je v IPv6 obdoba IPv4 multicastu. Funkce zůstala stejná jen s tím rozdílem, že takto lze kontaktovat pouze členy dané skupiny a pokud se v síti nevyskytuje člen dané skupiny, router jednoduše nic nepošle, tedy oproti broadcastům je zde méně prostoru pro útoky. Adresy individuální znamenají opět analogii k IPv4, ovšem je tu menší odlišnost, kdy existují lokální linkové adresy, po kterých lze také vést komunikaci, ale pouze v rámci LAN – nejsou routovatelné, tyto adresy se na zařízení přidělují automaticky po zapnutí podpory pro IPv6. A jak je již z tabulky patrné, největší prostor zabírají individuální globální adresy, což jsou stejné adresy jako veřejné IPv4 adresy – adresy propagované a dosažitelné v rámci celého internetu. Následuje příklad individuální globální adresy ve zkráceném a úplném tvaru.

2a01:490:18:fe0::1/64 = 2a01:0490:0018:0fe0:0000:0000:0000:0001/64

Úplnou novinkou u IPv6 jsou takzvané Anycast adresy, kdy tuto adresu má více uzlů najednou, ovšem komunikace od odesílatele se doručí pouze nejbližšímu uzlu, který má danou adresu. Její tvar je prefix:0:0:0:0, tedy pro naši vzorovou adresu by vypadala

takto: *2a01:490:18:fe0::* , v případě dotazu na tuto adresu dostaneme odpověď nejbližšího routeru dané podsítě.

11.3 ICMPv6

IPv6 verze ICMP, označována jako ICMPv6 prošla značnou revizí oproti staršímu formátu. Důvodem bylo především roztříštěnost diagnostických mechanismů mezi ICMP a ARP a také bezpečnostní hlediska, kdy se ICMP v řadě případů využívala k útokům typu (D)DoS[35] – cíl se zahltil velkým množstvím ICMP zpráv a téměř nic jiného nestíhal vyřídit, tak řada správců raději ICMP na svém stroji zakázala, což ovšem je v rozporu s příslušným RFC. Z tohoto důvodu obsahuje ICMPv6 bezpečnostní mechanismy, jako např. zabudovanou podporu autentizace / šifrování, kdy uzel může být nakonfigurován, aby přijímal pouze prověřené ICMPv6 zprávy, kvantitativní omezení, kdy správce je schopen omezit počet ICMPv6 zpráv na určitou pevně danou hladinu, a řadu dalších[36].

Zprávy ICMPv6 se opět dělí do dvou tříd:

- chybové – typ 0 až 127
- informační – typ 128 až 255

11.3.1 Objevování sousedů

I z pohledu diagnostických nástrojů je ICMPv6 je více komplexní než jeho předchůdce. Při jeho vytváření bylo myšleno na to, jak zlepšit a rozšířit vlastnosti ICMPv4, takže například objevování sousedů *(neighbor discovery)* - mechanismus, který nahrazuje ARP, pokrývá jeho původní schopnosti, ale přidává řadu zajímavých vlastností[37]:

- zjišťování linkových adres uzlů ve stejné lokální síti
- rychlá aktualizace neplatných položek a zjišťování změn v linkových adresách
- hledání směrovačů
- přesměrování

35 DDOS *(Distributed Denial of Service)* je typ útoku, který má za úkol zahltit cíl takovým způsobem, aby narušil provoz služeb, které poskytuje. Využívají se k tomu dost často domácí stanice napadené např. nějakým trojským koněm. Takovýmto počítačům vzdáleně používaným k útokům se říká Botnety.

36 SATRAPA PAVEL, IPv6, Internetový protokol IPv6, s. 91

37 SATRAPA PAVEL, IPv6, Internetový protokol IPv6, s. 97

- zjišťování prefixů, parametrů sítě a dalších údajů pro automatickou konfiguraci
- ověřování dosažitelnosti sousedů
- detekce duplicitních adres

Objevování sousedů přináší také úplnou novinku a tím je nový způsob automatické konfigurace nazývané jako bezstavová konfigurace na rozdíl od stavového DHCPv6. Rozdíl je jednoduchý, u stavové autokonfigurace si daný server udržuje přehled o tom, kdo jakou adresu aktuálně používá, komu jakou už přidělil, na jak dlouho, z jakého rozsahu může přidělovat dalším stanicím. Naopak bezstavová konfigurace v praxi znamená, že daný router v náhodných intervalech posílá ICMPv6 zprávy – ohlášení směrovače *(Router Advertisement, č. 134)*, které obsahují návod jakou IPv6 adresu si mají dané stanice přidělit a kdo je jejich implicitní směrovač a po jakou dobu. Lze ohlásit i více informací, jako například defaultní MTU dané sítě. Bezstavová konfigurace má ovšem zatím jednu zásadní nevýhodu a tou je nemožnost poslat stanicím i adresy DNS serverů pro danou síť. Na tomto problému se v současné době pracuje, aby se tak odstranil jediný náskok, který zatím drží DHCPv6[38].

Po přijetí tohoto ohlášení použije stanice modifikovaný formát MAC adresy daného síťového rozhraní EUI-64, zkusí poslat do sítě dotaz, jestli někdo danou IPv6 adresu již nepoužívá. V případě, že odpověď nedostane, přidělí si takto dopočtenou adresu, postup je zřejmý z následujícího příkladu, základ tvoří pozměněná MAC adresu, do které se vloží následujících 16 bitů – FF:FE a tím je IPv6 adresa kompletní.

MAC adresa: 00:23:54:37:8a:3e

IPv6 adresa: 2a01:490:18:3e00:223:54ff:fe37:8a3e/64

Mezi další ICMPv6 zprávy používané ND patří výzva směrovači (*router solicitation, č. 133)*, která se využívá k promptnímu kontaktování routeru z důvodu zjištění aktuální situace ze strany koncové stanice. A ICMPv6 zprávy výzva sousedovi *(neighbor solicitation, č. 135)* a ohlášení souseda *(neighbor advertisement, č. 136)*, se používají ke zjišťování linkové adresy na základě znalosti IP adresy[39]. Využívá se k tomu speciální skupinová adresa o společném prefixu:

ff02::1:ff00::/104

Kdy se pro dotaz použijí posledních 24 bitů známé IPv6 adresy, a tedy pro naši

38 SATRAPA PAVEL, IPv6, Internetový protokol IPv6, s. 97
39 SATRAPA PAVEL, IPv6, Internetový protokol IPv6, s. 97

vzorovou IPv6 adresu by měl dotaz následující tvar:

ff02::1:ff37:8a3e

Tím ovšem práce IPv6 nekončí, na rozdíl od IPv4 aktivně sleduje stav a dosažitelnost sousedů a to dvojím způsobem, buď za pomoci informací z některé z vyšších vrstev *(např. TCP)* či se jednoduše dotáže na stav opět pomocí výzvy sousedovi, pokud od něj dorazí ohlášení, je vše v pořádku[40].

11.4 Komunikace v IPv6 sítích

Z pohledu routování se oproti IPv4 téměř nic nemění, implicitní cesta je stále dána nulovým prefixem, u IPv6 tedy ::/0, následující přehled ukazuje základní routovací tabulku:

2a01:5f0:1:80::/64 dev eth0 proto kernel metric 256 expires 2592130sec mtu 1500 advmss 1440 hoplimit 0

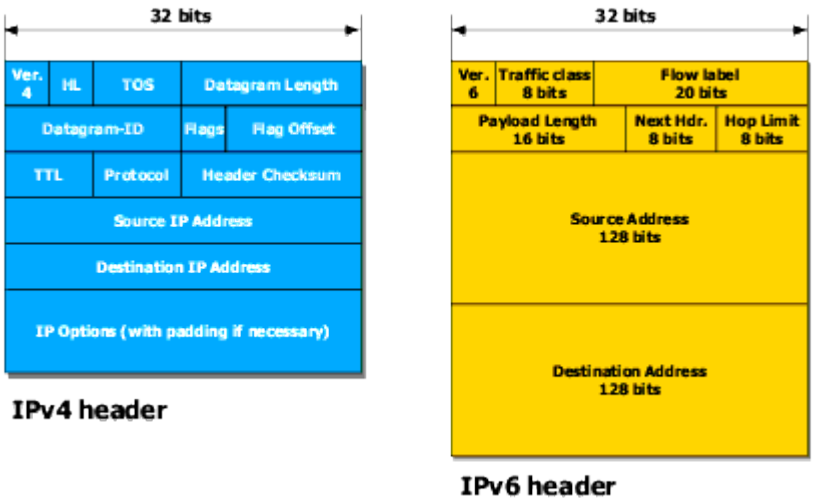
fe80::/64 dev eth0 proto kernel metric 256 mtu 1500 advmss 1440 hoplimit 0

default via fe80::230:48ff:fe9f:3143 dev eth0 proto kernel metric 1024 expires 1768sec mtu 1500 advmss 1440 hoplimit 64

V případě dynamického routování je nutné, aby daný protokol uměl pracovat s jiným typem adres, tedy IPv6 adresami. V kapitolách 3.3.3.1 a 3.3.3.2 je uvedeno, které routovací protokoly toto umožňují, případně v jaké jejich verzi byla tato podpora zahrnuta.

Hlavní rozdíl mezi IPv4 a IPv6 je ve velikosti hlavičky jejich datagramů. Ta má u IPv6 fixní velikost 40B a obsahuje pouze to nejnutnější. Co se do ní nevejde, je nutné začlenit formou rozšiřujících hlaviček. Oproti IPv4 pozbývá IPv6 hlavička především kontrolní součet, fragmentaci a tag rozšiřující volby[41].

40 SATRAPA PAVEL, IPv6, Internetový protokol IPv6, s. 97
41 PETR EMANUEL, Implementace IPv6, [online]



Obrázek 1: Porovnání IPv4 a IPv6 datagramů

Jak je z obrázku patrné, největší prostor základní hlavičky zabírá zdrojová a cílová adresa, stejně jako IPv4 obsahuje položku verze, jen zde má hodnotu 6. Dále tato hlavička obsahuje položku třídy provozu *(traffic class)*, která umožňuje zařadit pakety do určité QOS třídy *(quality of service)*, tedy priorizovat zpracování určitých paketů před ostatními. Zatím se ovšem téměř nevyužívá. Další méně využívanou položkou je značka toku *(flow label)*, která má v budoucnu posloužit k seřazení paketů podle určitých shodných vlastností (zdroj, cíl, požadavky na vlastnosti spojení, aj.). Délka dat *(payload lenght)* skrývá údaj o délce datagramu za hlavní hlavičkou, která se tedy do této hodnoty nezapočítává. Pro identifikaci následující rozšiřující hlavičky se používá položka další hlavička *(next header)*. Zbývá tedy už jen položka dosah *(hop limit)*, která nahrazuje položku životnosti datagramu - TTL *(time to live)* u IPv4 datagramů. Výhoda této základní hlavičky je nesporná, ušetří čas při zpracování datagramů po cestě k cíli[42].

Položka další hlavička umožňuje řetězit za sebe další hlavičky podle potřeby komunikace, například hlavička pro směrování, fragmentaci, šifrování obsahu (ESP), aj. Aby v hlavičkách a jejich zpracování nebyl chaos, je dáno jejich pořadí, což má opět pomoci v optimalizaci routování datagramů[43].

42 SATRAPA PAVEL, IPv6, Internetový protokol IPv6, s. 97
43 SATRAPA PAVEL, IPv6, Internetový protokol IPv6, s. 97

11.5 IPsec

Při návrhu IPv6 se též myslelo na bezpečnost, z tohoto důvodu je IPsec pevnou součástí IPv6. Má dva způsoby použití a to:

- **autentizaci** – kdy je cílem ověřit, zda odesílatel je skutečně ten, za kterého se vydává
- **šifrování** – umožňuje zašifrovat obsah komunikace[44]

Realizace obou služeb spočívá ve dvou rozšiřujících hlavičkách: AH *(autentication header)* a ESP *(encapsulation security payload)*, přičemž AH umí pouze autentizaci a ESP k tomu navíc umí i šifrovat obsah komunikace. Z tohoto důvodu se častěji využívá služeb ESP hlavičky, jejíž implementace je dle RFC oproti AH povinná[45].

Celá tato ochrana pracuje ve dvou režimech. V transportním režimu, kdy se rozšiřující hlavičky vkládají přímo do paketu, a nebo v tunelovacím režimu, kdy je stávající paket zabalen do nového paketu a opatřen hlavičkami novými včetně bezpečnostních. Praktický rozdíl je především v tom, že případný útočník u transportního režimu zná jak adresu odesílatele, tak adresu příjemce, naopak u tunelovacího režimu zná pouze adresu dvou bodů, mezi kterými probíhá šifrovaná komunikace a původní komunikace je chráněná. Z toho plyne, že je možné využívat bezpečnostních mechanismů nejen po celé trase komunikace, ale třeba jen na části. Například firma se dvěmi pobočkami má komunikaci uvnitř své sítě nešifrovanou, ale spojení mezi pobočkami již šifrované je[46].

44 SATRAPA PAVEL, IPv6, Internetový protokol IPv6, s. 189
45 SATRAPA PAVEL, IPv6, Internetový protokol IPv6, s. 189
46 SATRAPA PAVEL, IPv6, Internetový protokol IPv6, s. 189

12. Případová studie

Tato případová studie se týká routování páteřní sítě JM-Net o.s. za pomoci GNU/Linux, konkrétně distribuce Debian GNU/Linux. Výběrem vhodného hardware, instalací daných routerů se tato případová studie zabývat nebude. Zaměří se pouze na důležité aspekty související s routováním včetně výběru vhodného software, se zaměřením především na routování nového protokolu IPv6.

Operační systém GNU/Linux, v tomto případě distribuce Debian, je jako taková velice dobře připravena plnit úlohu routeru, ostatně řada menších providerů v rámci českého NIXu s nimi routuje celou svou síť (AS). Distribuce Debian byla zvolena, protože se jedná o jednu z nejrozšířenějších „free“ distribucí, která ve své stabilní verzi má dlouhodobou podporu v podobě oprav chyb a konzistentnosti softwareového vybavení, navíc v rámci JM-Net o.s. je několik členů, kteří s ním dokáží bez problémů pracovat.

Síť JM-Net o.s. se skládá z optické páteřní sítě, routovanou za pomoci směrovačů s operačním systémem GNU/Linux, které se věnuje tato případová studie, a wifi části, která je routovaná kombinovaně buď za pomoci proprietárního software Mikrotik nebo různých minimalistických distribucí, nejčastěji Voyage Linux. V rámci sítě se pro routování využívá dynamického routování a to konkrétně routovacího protokolu OSPF. Pro komunikaci s vnějším prostředím je použit protokol BGP.

Při výběru routovacího software je tedy z důvodu předchozích požadavků, na výběr hlavně routovací balík Quagga a také BIRD z produkce českého registrátora .cz domény - CZ.NIC. Oba dva dokaží routovat pomocí BGP či OSPF jak IPv4 tak IPv6, ovšem z důvodu známějšího stylu konfigurace, podobné směrovačům Cisco, byla zvolena Quagga.

12.1 Popis páteřní sítě JM-Net o.s.

Z důvodu upgradu páteřní sítě z duálních 5GHz Wi-Fi spojů za pronajatá optická vlákna, nákupu gbit optických převodníků, změně poskytovatele zahraniční konektivity poskytujícího i IPv6 peering bylo nutné změnit stávající systém routování páteřní sítě, kterou zatím routoval slabší hardware ve formě platforem např. routerboard či wrap, které nemají dostatečný výkon.

Z cenově dostupných řešení bylo nakonec vybráno to, které je založeno na platformě IBM PC, konkrétně výrobce Intel s procesory Intel Core 2 Duo s využitím software Debian GNU/Linux. Takto výkonný hardware je již schopen daleko lépe využít kapacity Gbit optické linky a není tak již úzkým hrdlem, jak tomu bylo u platforem Wrap a Routerboard.

Sdružení JM-Net o.s. tedy posílilo každý přístupový bod do páteřní sítě o samostatný router založený na platformě Intel Core 2 Duo. Pro páteřní síť též byla vyčleněna speciální VLAN s vlastním neveřejným rozsahem. Všechny body z páteřní sítě ústí do agregačního switche Juniper EX2200, který funguje jako L2 prvek pro celou optickou část sítě. Optické páteřní linky mají fyzické vyústění v datovém centru Sitel, kde je nainstalován hraniční router, který zprostředkovává komunikaci s vnějším prostředím. Příloha C obsahuje obrázek topologie této páteřní sítě.

V rámci sítě JM-Net o.s. panují určitá historická pravidla, jde o členskou síť celorepublikové neveřejné CZFree.net sítě a jako taková má přidělený neveřejný rozsah *10.38.0.0/16*, se kterým lze samovolně nakládat – s ohledem na dodržení obecných doporučení RFC. Adresní plán je vcelku jednoduchý a to takový, že jednotlivým přístupovým bodům číslovaným *1-254* připadá jeden síťový loopback z rozsahu *10.38.0.X/32* a jeden rozsah z *10.38.X.0/24*, kde X je číslo přístupového bodu, číslováno chronologicky, podle data vzniku.

12.2 Základní nastavení routování v OS Linux

Pokud máme čerstvě nainstalovaný systém, je třeba nastavit základní síťové nastavení a zapnout forwarding paketů v jádře, v distribuci Debian k síťovému nastavení slouží soubor */etc/network/interfaces*, jeho obsah vypadá přibližně takto:

```
# The loopback interface
auto lo
iface lo inet loopback
auto eth0
iface eth0 inet static
address 10.38.4.33
netmask 255.255.255.248
broadcast 10.38.4.39
```

Do tohoto souboru je ještě dobré zavést speciální interface *dummy0*, který bude fungovat jako naroutovaný síťový loopback odpovídající router-id v OSPF. V praxi to

znamená, že každý router bude mít vlastní unikátní IP adresu z konkrétního neveřejného rozsahu *10.38.0.0/24* a pod tím bude dosažitelný v rámci celé czfree.net sítě a také tento interface bude vždy aktivní a IP adresa na něm nastavená bude dosažitelná i v případě, že síťové interface budou shozené, což zapříčiní nedosažitelnost IP adresy na ní nastavené. Toto se hodí například při monitoringu routerů na síti, kde se vyhodnocuje dosažitelnost routeru jako takového a ne jeho síťových rozhraní.

```
auto dummy0
iface dummy0 inet static
address 10.38.0.4
netmask 255.255.255.255
broadcast 10.38.0.4
```

Interface *dummy0* je v tomto případě vhodný i k použití routované veřejné IP adresy, která odpovídá opět číslu routeru a je z rozsahu *78.108.106.X/32*, zadává se takto[47]:

```
up /sbin/ip ad ad 78.108.106.4/32 dev dummy0
```

Pokud je takto IP adresa zadána, objeví se rozdílný výstup v příkazech *ifconfig* a *ip a*. i*fconfig*, který je zastaralý, totiž nezobrazí tímto způsobem přidané alias IP adresy na daných zařízeních, proto je vhodnější používat pouze příkaz *ip.*

Po tomto kroku máme základní sadu IP adres zanesenou do systému a takto nastavené nezmizí ani po restartu routeru. Můžeme tedy přejít k dalšímu kroku a to je povolení forwardingu v jádře Linuxu, tento krok je důležitý pro fungování Linuxu jako routeru. V Debianu se nejlépe upraví soubor */etc/sysctl.conf* a to tak, že odkomentujeme - smažeme # na začátku dané řádky, aby v souboru existoval řádek s tímto nastavením:

```
net.ipv4.ip_forward=1
```

či povolíme ihned, ale pro stálé nastavení je nutné upravit výše uvedený soubor:

```
sysctl net.ipv4.ip_forward=1
```

Takto máme již router připraven pro předávání paketů z jednoho interface na druhý, což je to, co požadujeme a je hlavní činností routeru.

Zde je vhodné zmínit ještě situaci, kdy router neobsluhuje tak velkou síť, respektive za tímto routerem nefungují další routery a obsluhuje např. jednu vnitřní síť, typicky LAN síť v domácnosti či v kanceláři případně malé firmě. Zde není třeba instalovat démona, který zvládá dynamické routování, ale lze si bez problému vystačit

47 NEMETH E., SNYDER G., HEIN T. R., Linux, Kompletní příručka administrátora, s. 297

se staticky řešeným routováním.

Tyto statické routy je nutné zadávat ručně a k tomu v Linuxu slouží příkaz *ip*:

ip route add 10.38.254.0/24 dev eth0

a pro zadání defaultní routy:

ip route add default via 10.38.254.254

12.3 Dynamické routování

V případě rozlehlé sítě není skoro možné udržovat routovací tabulku na každém routeru uvnitř sítě ručně, navíc dynamické routování přináší hlavní výhodu v tom, že reaguje na změny v síti. Vzhledem k požadavkům byl vybrán routovací balík Quagga. Tento balík obsahuje několik démonů rozdělených podle jednotlivých routovacích protokolů – ospfd, ospf6d, ripd, ripngd, bgpd a centrálního démona pojmenovaného z historických důvodů zebra. Z tohoto výčtu využijeme pouze tučně vypsané démony.

Konfigurace Quaggy je možná více způsoby, je možné editovat konfigurační soubory přímo ručně s tím, že při každé změně je nutné restartovat celého démona:

#/etc/init.d/quagga restart

Tento způsob není moc šikovný pro prvotní konfiguraci, ovšem v případě, že máme již hotovou funkční konfigurací a potřebujeme ji přenést na jiný router, tak ušetříme spoustu času. Další způsob konfigurace je možný pomocí telnetu a to připojením se na port, na kterém daný démon naslouchá, pro centrálního zebra démona takto:

 # telnet localhost 2601

Kde po zadání hesla máme přístupnou konzoli a můžeme zadávat příkazy ve stylu konfigurace směrovačů Cisco, které se ihned projeví v chování routeru. Tento způsob se nazývá interaktivní. Ovšem i tento způsob má velkou nevýhodu, pro konfiguraci je nutné přeskakovat mezi jednotlivými démony, proto byl vynalezen jednotný terminál jménem *vtysh*, který sjednocuje nastavování jednotlivých démonu do jednoho centrálního prostředí a konfigurace je opět v interaktivním režimu jako u směrovačů Cisco. Tento způsob bude použit v této praktické části.

12.4 Instalace a prvotní nastavení Quaggy

Distribuce Debian standardně obsahuje tento balík a tak není problém ho

nainstalovat jednoduchým příkazem:

aptitude update && aptitude install quagga

Po nainstalování se můžeme pustit do prvotního nastavení. Nejlépe je začít zkopírováním vzorových konfiguračních souborů:

#cp /usr/share/doc/quagga/examples/zebra.conf.sample /etc/quagga/zebra.conf
#cp /usr/share/doc/quagga/examples/ospfd.conf.sample /etc/quagga/ospfd.conf

A jelikož ke konfiguraci budeme používat *vtysh* tak i:

#cp /usr/share/doc/quagga/examples/vtysh.conf.sample /etc/quagga/vtysh.conf

Zde je dobré před dalším postupem zkontrolovat zakomentování a případně přidat na počátek řádku „!“, tak aby řádek vypadal následovně:

!service integrated-vtysh-config

Toto by v nezakomentovaném stavu způsobilo, že celá konfigurace Quaggy bude při ukládání aktuální konfigurace zapsána do jednoho souboru, což přehlednosti moc nepomůže, spíše naopak. Navíc v oficiální dokumentaci Quaggy je doporučeno držet se jednoho stylu. My se tedy budeme držet konfigurace v oddělených souborech.

Po tomto je ještě třeba změnit práva:

#chown quagga:quaggavty /etc/quagga/.conf*
#chmod 640 /etc/quagga/.conf*

Dále přidat následující nastavení, aby *vtysh* nekončil po každém příkazu zaseknutím a čekáním na stisk klávesy *q*.

#echo VTYSH_PAGER=more > /etc/environment

A nakonec zrestartovat Quaggu, aby si načetla základní konfiguraci.

#/etc/init.d/quagga restart

Po odhlášení a přihlášení máme Quaggu připravenou k použití[48].

12.5 Konfigurace součástí Quaggy

Instalaci jsme tedy provedli, můžeme se pustit do nastavení, pomocí příkazu *vtysh* se přihlásíme do příkazového řádku Quaggy. Toto rozhraní má více módů, tento po přihlášení je *prohlížecí mód*, kde příkazem *show ?* vypíšeme možnosti, které lze zobrazit. Například můžeme provést kontrolu zapnutého forwardování paketů:

48 QUAGGA – The easy tutorial – How to use Quagga, [online]

j*izak.jiznak.czf# show ip forwarding*

IP forwarding is on

Pokud se chceme přepnout do editačního módu, musíme použít příkaz:

jizak.jiznak.czf# configure terminal

jizak.jiznak.czf(config)#

Nyní můžeme zadávat jednotlivé příkazy a ty se ihned projeví v chování routeru. Začneme hlavním démonem jménem *zebra.* Tento démon slouží především k nastavení statických rout, nastavení obecného chování – hesla, logování, atd. a také k různým doplňkovým službám jako je router-advertisement u IPv6, ale neobsluhuje žádný konkrétní routovací protokol. Konfiguraci začneme po přihlášení se do vtysh pomocí příkazu *configure terminal*, kdy se na řádku objeví:

jizak.jiznak.czf(config)#

Tento výstup znamená, že můžeme zadávat jednotlivé příkazy. Vhledem k topologii sítě by konfigurace démona zebra měla vypadat přibližně takto, příkazy jsou kurzívou, komentáře normálním fontem:

! Zebra configuration saved from vty
! 2011/03/12 16:55:30
hostname jizak – jméno routeru
password tajneheslo – heslo pro privilegovaný režim
log file /var/log/quagga/ospf.log – do jakého souboru se bude logovat výstup
log record-priority – zapíše do logu také závažnost logovaného údaje
!
interface dummy0 – jednotlivý interface
ipv6 nd suppress-ra – zakáže posílání oznámení směrovače (IPv6) pro daný interface
!
...
!
ip route 10.38.21.112/28 10.38.4.221 – takto se zadává statická routa
ip route 10.38.249.0/27 10.38.4.216
ip route 78.108.106.20/32 10.38.4.221
ip route 78.108.106.108/32 10.38.4.221
access-list term deny any – restrikce přístupu do konfiguračního rozhraní démona
!
router-id 10.38.0.4 – unikátní číselné označení routeru v síti
ip forwarding – zapnuté forwardování paketů
!
line vty[49]

49 Kompletní znění v příloze B

Tento výstup získáme provedením příkazu *show running-config*, je dobré příkazy zadávat jednotlivě či maximálně po menších blocích a kontrolovat aktuální konfiguraci uvnitř konfiguračního módu pomocí příkazu *do show running-config*, který umožní spouštět příkazy, které jsou určené pro jiný než tento mód. Po zadaní všech následujících příkazů je nutné, tak jako na Cisco směrovačích, konfiguraci uložit do stálé konfigurace, což provedeme příkazem:

```
jizak.jiznak.czf# write
Building Configuration...
Configuration saved to /etc/quagga/zebra.conf
[OK]
```

Mezi další zajímavá nastavení, která ale nejsou povinná, patří:

service advanced-vty – zapne pokročilejší funkce vty

service password-encryption – zapne kryptování hesel v konfiguračních souborech

log file /dev/null – po odladění chyb je vhodné logovat do ztracena, log jinak znatelně narůstá

ip ospf authentication-key AUTH_KEY – zabezpečí OSPF pakety jednoduchým klíčem, nastavuje se k určitému síťovému rozhraní a je nutné ho zadat na obou stranách spojení, tj. mezi jednotlivými sousedy.

ip ospf message-digest-key KEYID md5 KEY - podobné jako předchozí volba, jen s rozdílem použití bezpečnější MD5 šifry, KEYID značí číslo md5 klíče a po md5 následuje již vlastní klíč, vzhledem k typu šifry je nutné mít na daném routeru aktuální čas, nejlépe aktualizovaný pomocí NTP protokolu

Nyní již pokročíme ke konfiguraci OSPF démona, následuje příklad základní konfigurace tohoto démona, kterého je nejdříve nutné zapnout v */etc/quagga/daemons* pouze změnou z *no* na *yes* a po té restartovat Quaggu:

```
# /etc/init.d/quagga restart
```

Poté se již můžeme pustit do konfigurace OSPF démona, opět pomocí *vtysh* → *configure terminal ->router ospf.* Zde již můžeme zadávat následující údaje:

```
! Zebra configuration saved from vty
!   2011/03/13 17:59:49
!
hostname jizak
password ahoj
log file /var/log/quagga/ospf.log
!
```

...
!
interface eth1
 ip ospf cost 1 – cena cesty specifická pro tento konkrétní interface
!
...
!
router ospf
 ospf router-id 10.38.0.4
 redistribute connected metric-type 1 – zapnutí distribuce připojených
 redistribute static metric-type 1 – zapnutí distribuce statických rout a s jakou metrikou
 passive-interface eth1 – do passive se neposílají OSPF pakety
 passive-interface eth2
 network 10.38.0.4/32 area 10.38.0.0 – specifikace sítě do určité arey
 ...
 network 78.108.106.244/32 area 10.38.0.0
!
!
line vty
 access-class term
!

Tímto máme nakonfigurovaný OSPF router pro routování uvnitř sítě *(AS)*, pokud ale chceme být dostupní pro zbytek světa *(Internetu)*, je nutné zapojit do hry i hlavní router a použít BGP protokol.

Na hlavním router, v našem případě router Sitel, je nutné napřed zkontrolovat, zda má Quagga démona BGP zapnutého, opět tedy */etc/quagga/daemons*, konfiguraci BGP pustíme v konfiguračním rozhraní *vtysh*, za pomoci příkazu *router bgp*[50], následuje přehled nejdůležitějších voleb:

router bgp 64538 – přidělené číslo AS
bgp router-id 10.253.32.15 – numerické ID routeru
 ip prefix-list PUB description Public IP – popis prefix listu – filtru adres
ip prefix-list PUB seq 10 permit 78.108.106.0/24 – jaké adresy jsou v rámci prefix-listu povoleny
ip prefix-list PUB seq 11 permit 212.79.108.0/24
ip prefix-list PUB seq 99 deny any – vše od této sekvence dál je zakázané
neighbor NFXPUB peer-group
 neighbor NFXPUB remote-as 8251 – k jakému AS patři daný soused
 neighbor NFXPUB description NFX public route-servers – popis souseda
 neighbor NFXPUB next-hop-self – router oznamuje svou adresu jako next-hop cestu
 neighbor NFXPUB send-community both – vzájemné zasílání komunit
 neighbor NFXPUB soft-reconfiguration inbound
 neighbor NFXPUB prefix-list PUB out – použitý adresný filtr

50 Kompletní konfiguraci obsahuje příloha A

V tuto chvíli máme hotovou konfiguraci routování pro IPv4 adresy. Toto jsou ovšem pouze základní možnosti nastavení. Lze například rozdělit sítě za těmito routery do jiných oblastí *(areas)* z důvodu méně náročné výměny routovacích informací mezi routery.

12.6 Specifika routování IPv6

Routování sítě po IPv4 máme funkční, můžeme tedy přikročit ke konfiguraci IPv6. V současné době je asi nejvýhodnější stavět podporu pro IPv6 pouze jako doplněk k IPv4 síti, stavět síť podporující pouze IPv6 není momentálně z hlediska služeb fungujících na IPv6 moc přínosné. Z tohoto důvodu má většina poskytovatelů síť postavenou jako dual-stack, kde odděleně probíhá komunikace jak po IPv4 tak po IPv6, to bude i tento případ. Momentálně není např. možné bez různých pomůcek – tunely, připojit koncové počítače, které mají v cestě domácí router se zapnutou funkcí NAT a nezavedenou podporou IPv6, proto se může topologie IPv4 a IPv6 sítí lišit.

12.6.1 Adresní plán IPv6

Jak bylo již představeno v kapitole Routování IPv6, nový protokol používá zcela odlišný systém zápisu adres oproti staršímu standardu IPv4. Vzhledem k tomu, že většina sítí se momentálně staví jako dual-stack sítě, je třeba vymyslet určitou analogii při adresování stávající infrastruktury IPv6 adresami. Tento problém řeší předem připravený adresní plán. V JM-Netu, jak již bylo popsáno v kapitole 5.1, používáme číselné označení jednotlivých routerů chronologicky podle data vzniku a každý takovýto router má přidělený /24 IPv4 adresní prostor. Tento systém, ač se může zdát jako plýtvání, tak funguje a administrátorům poskytuje rychlou orientaci v síti a také určitou volnost, protože adres je pro daný přístupový bod dostatek a pro opravdu velké přístupové body s přehledem stačí. Také IP adresy pro spoje se z důvodu velkého prostoru berou z přiděleného subnetu o velikosti /24 (256 adres). Toto je jen základní výčet požadavků na adresní plán, ale bylo nutné je do daného IPv6 adresního plánu zapracovat.

Pro IPv6 adresní plán byl tedy nakonec zvolen systém s co nejbližší analogii ke stávajícímu IPv4 plánu a zároveň poskytující správci daného přístupového bodu určitou volnost. JM-Net má od svého LIR, kterým je NFX z.s.p.o., přidělen rozsah

2a01:490:/48. Jednotlivý přístupový bod má přidělen 2a01:490:c1:XXYY/56, s tím, že XX znamená číslo přístupového bodu v hexadecimálním tvaru a YY je číslo jednotlivých síťových rozhraní na daném přístupovém bodu. Každý přístupový bod má tedy možnost přidělit IPv6 adresy o doporučené délce /64 na 256 síťových rozhraní a zároveň routerů může být až 256. Tento systém vyhovuje požadavkům na něj kladených, ovšem opět podporuje plýtvání adresami, což v konečném důsledku je též vlastnost celého IPv6 protokolu. Tento adresní plán ovšem vydrží na rozumně dlouhou dobu – odhadem na příštích 5 let. V případě dosažení počtu 256 přístupových bodů již bude pro JM-Net možné stát se LIR v rámci RIPE nebo si nechat přidělit další /48 rozsah.

12.6.2 Routování IPv6 s Quaggou

Adresní plán již máme hotový, můžeme se tedy opět pustit do konfigurace Quaggy, z důvodu jiného protokolu existuje i jiný démon Quaggy pro IPv6 – ospf6d. Funkce má podobné jako ospfd, ovšem z důvodu začlenění IPsec do IPv6, tak tento démon již neřeší autentizaci jako ospfd, ale je řešena na úrovni Ipv6 (Ipsec).

Pro povolení forwardování IPv6 paketů je třeba nejprve odkomentovat následující volbu v */etc/sysctl.conf*:

net.ipv6.conf.all.forwarding=1

či pro okamžité nastavení:

sysctl net.ipv6.conf.all.forwarding=1

Nyní již můžeme přistoupit k přímé konfiguraci IPv6 pomocí *vtysh*, případně nejdříve povolíme v */etc/quagga/daemons* démona ospf6d, pro konfiguraci slouží příkaz *router ospf6*:

!
! Zebra configuration saved from vty
! 2011/03/13 17:59:49
!
hostname jizak
password router
enable password vladqa89
log file /var/log/quagga/ospf.log
!
debug ospf6 lsa unknown – zapnutý debug ospf6
!
interface eth0

...
!
interface eth1
ipv6 ospf6 cost 1 – cena cesty tohot konkrétního síťového rozhraní
ipv6 ospf6 hello-interval 10 – interval posílání hello paketů
ipv6 ospf6 dead-interval 40 – kolik sekund musí uplynout od neobdržení hello packetů do prohlášení souseda za mrtvého
ipv6 ospf6 retransmit-interval 5 – čas mezi zasláním LSA paketů
ipv6 ospf6 priority 1 – router s vyšší prioritou je preferovanější
ipv6 ospf6 transmit-delay 1 – odhadovaný čas zaslání LSA paketů
ipv6 ospf6 instance-id 0
ipv6 ospf6 passive – OSPF neposílá do takového rozhraní oznámení
!
interface eth2
...
!
router ospf6
router-id 10.38.0.4
redistribute connected - povolí distribucí připojených (tj. vlastních) rout skrze OSPFv3
interface eth0 area 0.0.0.0
!
line vty
![51]

Pokud máme na nějaké síťové rozhraní, kde jsou přímo připojené *(L2 segment)* uživatelské stanice, je vhodné jim také IPv6 zpřístupnit k používání. Zřejmě nejjednodušší je využít služeb ohlášení směrovače a tedy bezstavové konfigurace.

Následující volba se zadává k jednotlivým síťovým rozhraním:

interface eth1
ipv6 address 2a01:490:18:fe02::1/64 – adresa daného síťového rozhraní
ipv6 nd ra-interval 150 – maximální počet sekund pro posílání RA oznámení
ipv6 nd prefix 2a01:490:18:fe02::/64 – oznamovaný prefix
no ipv6 nd suppress-ra – zruší zákaz posílání oznámení směrovače = povolí RA

Tímto máme nastavené routovaní IPv6 v rámci vnitřní sítě, ještě je nutné tento rozsah zpropagovat světu na bráně sítě – BGP routeru, opět tedy *vtysh → configure terminal → router bgp*:

router bgp 64538
neighbor 2a01:490:0:1::1 remote-as 8251
no neighbor 2a01:490:0:1::1 activate
neighbor 2a01:490:0:1::b:1 remote-as 8251
no neighbor 2a01:490:0:1::b:1 activate
address-family ipv6
network 2a01:490:18::/48

51 Kompletní konfigurace v příloze B

neighbor 2a01:490:0:1::1 activate
neighbor 2a01:490:0:1::1 send-community both
neighbor 2a01:490:0:1::1 soft-reconfiguration inbound
neighbor 2a01:490:0:1::b:1 activate
neighbor 2a01:490:0:1::b:1 send-community both
neighbor 2a01:490:0:1::b:1 soft-reconfiguration inbound
exit-address-family

Toto nastavení se po provedení příkazu *write*, přidá k již stávajícímu nastavení pro IPv4, tímto máme vše co se týká dynamického routování hotové.

Může se stát, že přes BGP přijde více rout, než v základu Linux zvládne *(4096 rout)* . Toto se řeší zvednutím následující položky na dostatečnou hodnotu:

sysctl -w net.ipv6.route.max_size=16384

12.6.3 Ověření funkčnosti IPv6

Po skončení konfigurace je ještě dobré ověřit funkčnost. Kontrolu stavu BGP provedeme pomocí: *show ipv6 bgp summary* případně *show ip bgp summary*, obojí zobrazí základní shrnutí stavu BGP routeru v podobě sousedství a počtu rout v lokální bázi.

OSPF lze zkontrolovat pomocí *show ip ospf neighbor*, které vypíše stav OSPF sousedů a případně pro zobrazení předaných rout: *show ip ospf route.* Případně můžeme zkusit průchod sítí pomocí traceroute6:

jizak:~# traceroute6 ipv6.google.com
traceroute to ipv6.google.com (2a00:1450:8007::68), 30 hops max, 40 byte packets
1 2a01:490:18:fe00::1 (2a01:490:18:fe00::1) 0.433 ms 0.409 ms 0.389 ms
2 l3sw-nfx1.v6.nfx.cz (2a01:490:0:1::1) 1.092 ms 1.074 ms 1.328 ms
3 nixcz-v6.net.google.com (2001:7f8:14::1d:1) 1.017 ms 1.010 ms 1.258 ms
4 2001:4860::1:0:10 (2001:4860::1:0:10) 10.397 ms 10.614 ms 10.597 ms
5 2001:4860::2:0:48d (2001:4860::2:0:48d) 10.579 ms 10.566 ms 10.540 ms
6 2001:4860:0:1::c9 (2001:4860:0:1::c9) 10.526 ms 2001:4860:0:1::c7 (2001:4860:0:1::c7) 10.664 ms 2001:4860:0:1::c9 (2001:4860:0:1::c9) 15.758 ms
7 2a00:1450:8007::68 (2a00:1450:8007::68) 8.934 ms 8.925 ms 8.918 ms

Takovýto výstup znamená, že máme IPv6 funkční. Možností je ale více, takže poslední příklad – zobrazení nalezených sousedů:

jmsitel:~# ip -6 neigh
fe80::225:90ff:fe31:414b dev eth1 lladdr 00:25:90:31:41:4b STALE

fe80::20c:42ff:fe21:8982 dev eth1 lladdr 00:0c:42:21:89:82 router STALE
fe80::20c:42ff:fe3b:41f3 dev eth1 lladdr 00:0c:42:3b:41:f3 router REACHABLE
...
2a01:490:0:1::b:1 dev eth0.4001 lladdr 00:22:56:b9:00:ff router REACHABLE

12.6.4 Zkušenosti s IPv6 po roce provozu

V rámci sítě JM-Net o.s. se IPv6 provozuje již více než rok, za tuto dobu se změnila řada věcí, především podpora tohoto protokolu ze strany výrobců síťových prvků, kdy již není problém narazit na domácí WiFi/LAN routery označené podporou IPv6. Tím odpadl snad poslední problém s dosahem IPv6 až ke koncovým uživatelům, kdy poskytovatel internetu sice na své páteřní síti IPv6 podporuje a provozuje, ale uživatel se k IPv6 nemá jak dostat, protože jeho domácí router tento protokol ignoruje.

Trochu se též zlepšila informovanost lidí. Řada lidí si chce nový protokol vyzkoušet. Celoplošné televize již v hlavním čase daly tématu přechodu na IPv6 prostor již několikrát. A webových portálů zabývající se tímto tématem je velké množství.

Z pohledu JM-Netu bude v následujícím roce cíl rozšířit IPv6 i na ten nejvzdálenější router v síti a zároveň tak i mezi připojené uživatele. Větší část sítě, především tedy její WiFi část je routovaná na Linuxu založeným Mikrotik OS a zde se za minulý rok objevila řada problémů ohledně podpory IPv6, kdy bylo nutné držet starší verzi tohoto routovacího OS, aby IPv6 šla použít. Tyto doby jsou naštěstí pryč a několik novějších verzí Mikrotik OS již prokázalo, že IPv6 podporu mají stabilní a vhodnou pro reálné nasazení.

Další oblastí, které bude nutné věnovat pozornost je zavedení použitých IPv6 adres do systému DNS, tak aby je bylo možné jednoznačně a zapamatovatelně identifikovat, minulý měsíc JM-Net o.s. prostřednictvím svých statutárních zástupců zažádal o delegování reverzní zóny pro svů rozsah *2a01:490:18::/48* a žádosti bylo vyhověno. Cílem je tedy dokončit podporu IPv6 na všech prvcích v síti a samozřejmě začít provozovat všechny služby na obou protokolech.

13. Závěr

Nový standard IPv6 neklade o moc větší požadavky na administrátora oproti stávajícímu IPv4. Jedinou překážkou tak může být jiný formát adresy a trochu jiné principy chování stanic v IPv6 sítích. Pro normálního uživatele IPv6 přináší řadu výhod. Především tedy to, že podporuje trend dnešní doby – přijít s přenosným počítačem, připojit se na bezdrátovou síť a pracovat – *plug and play*. Normální uživatel tedy nemusí vůbec postřehnout a asi často ani nepostřehne, že je jeho komunikace posílaná skrze nový protokol. V možnostech správce sítě je pak uživateli takový komfort poskytnout.

V praktické části, kapitole Případová studie, bylo prokázáno, že implementace IPv6 do již fungujícího prostředí je nenáročná. Nejdůležitější je mít předem jasný a daný adresní plán upravený podle konkrétních požadavků sítě a toho se při implementaci IPv6 držet. Není ani vhodné se příliš omezovat, adres je opravdu dostatečné množství. Linux jako takový prokázal, že je připraven routovat IPv6 protokol již hned po instalaci a po doinstalování dodatečného software zvládne routovat i větší sítě. Z jeho strany nepanují téměř žádná omezení, omezení je pouze na straně výkonu použitého hardware. Zatím je náročnost IPv6 na hardware o mnoho nižší z důvodu menších datových toků a také daleko nižšímu počtu rout, které si mezi sebou BGP routery předávají, časem se tedy teprve ukáže, jak dalece je nový protokol optimalizován pro rychlejší zpracování datagramů routerem.

V příštích letech se teprve ukáže nakolik jsou provideři připraveni na tento nový protokol. Autor zastává názor, že ač někteří zaspali, z důvodu nenáročnosti implementace, není problém náskok ostatních dohnat. Již téměř všichni výrobci síťových prvků mají u svých výrobků v základu zapnutou podporu IPv6. Autor tedy do budoucna předpokládá, že ač nyní je nástup IPv6 opatrný, brzy můžeme očekávat masivnější nástup.

14. Seznam literatury

SATRAPA PAVEL, IPv6, Internetový protokol IPv6, 2. vydání, Praha: CZ.NIC, 2008, 357 s., ISBN 978-80-904248-0-7

NEMETH E., SNYDER G., HEIN T. R., Linux, Kompletní příručka administrátora, 2. aktualizované vydání, Brno: Computer Press, 2008, ISBN 978-80-251-2410-9

DOSTÁLEK LIBOR, KABELOVÁ ALENA, Velký průvodce protokoly TCP/IP a systémem DNS, 2. aktualizované vydání, Praha: Computer Press, 2000, ISBN 80-7226-323-4

BOUŠKA PETR, Adresování v IP sítích, 21.7.2010, [online] <http://www.samuraj-cz.com/clanek/adresovani-v-ip-sitich/>

BOUŠKA PETR, Víte jak pracuje router?, 23.6.2010, [online] <http://www.samuraj-cz.com/clanek/vite-jak-pracuje-router/>

BOUŠKA PETR, TCP/IP – Routing - směrování, 21.9.2007, [online] <http://www.samuraj-cz.com/clanek/tcpip-routing-smerovani/>

BOUŠKA PETR, Cisco Routing 6 – srovnání routovacích protokolů, 28.4.2009, [online] <http://www.samuraj-cz.com/clanek/cisco-routing-6-srovnani-routovacich-protokolu/>

BOUŠKA PETR, Cisco Routing 6 – BGP – Border gateway protocol, [online], 18.4.2009, <http://www.samuraj-cz.com/clanek/cisco-routing-5-bgp-border-gateway-protocol/>

BARTOŠEK MIROSLAV, Krátce z historie Internetu, 1995, ISSN 1212-0901, [online] <http://www.ics.muni.cz/zpravodaj/articles/22.html>

KUNIHIRO ISHIGURO, et al., 1995-2005, [online] <http://www.quagga.net/docs/quagga.html>

PETR EMANUEL, Implementace IPv6, 2011, [online] <http://www.nic.cz/akademie/course/18/detail/>

QUAGGA – The easy tutorial – How to use Quagga, 5.7.2010, [online] <http://openmaniak.com/quagga_tutorial.php>

15. Seznam obrázků

16. Seznam tabulek

17. Přílohy

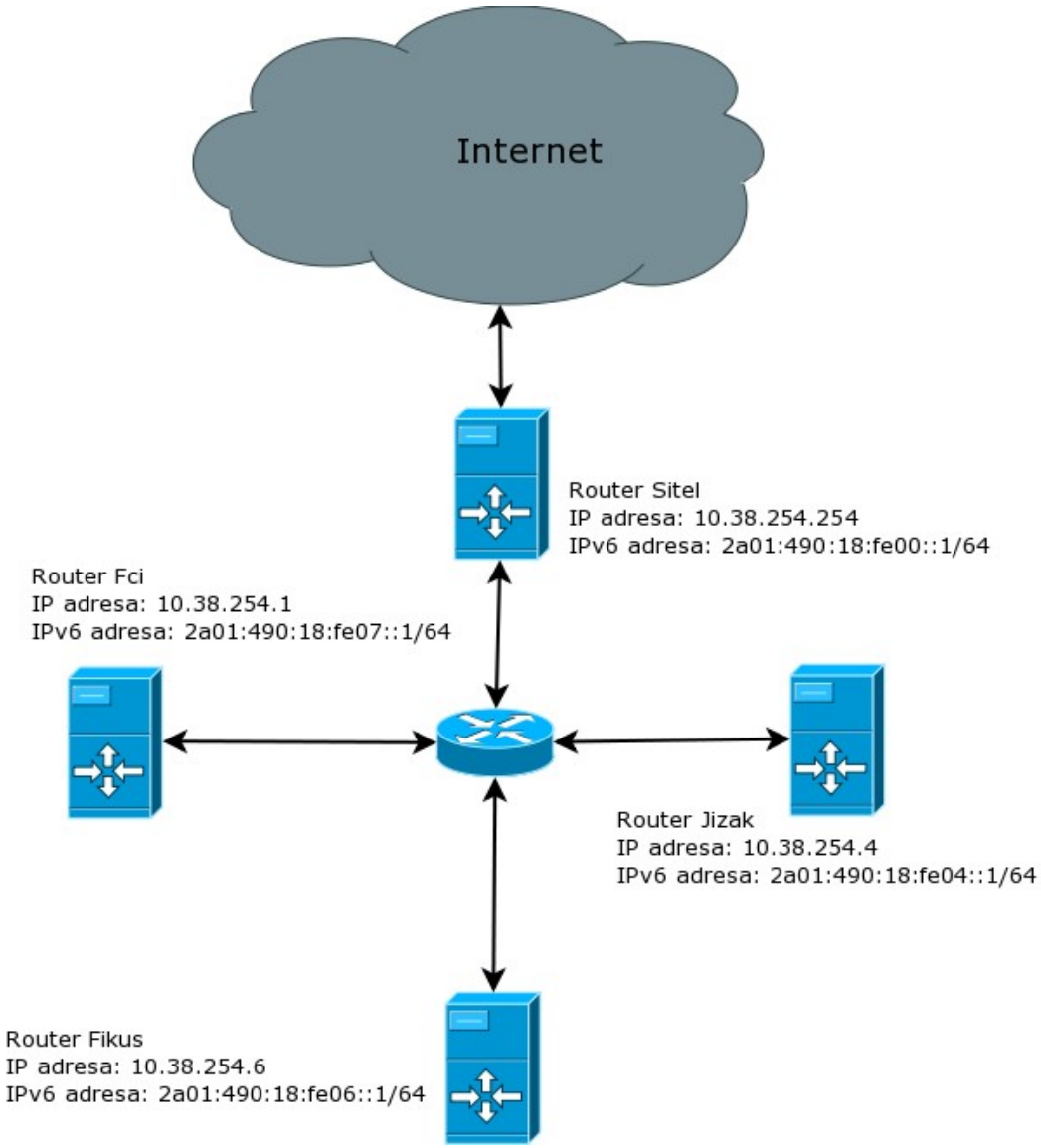
```
!
! Zebra configuration saved from vty
!   2011/01/17 12:39:41
!
hostname jmsitel
password tajne
enable password tajneheslo
log trap errors
log file /var/log/quagga/bgpd.log
log record-priority
service advanced-vty
service terminal-length 23
!
bgp config-type cisco
!
router bgp 64538
 no synchronization
 bgp router-id 10.253.32.15
 network 10.38.0.0 mask 255.255.0.0
 network 78.108.106.0 mask 255.255.255.0
 network 212.79.108.0 mask 255.255.255.0
 neighbor EXTERNAL peer-group
 neighbor EXTERNAL description Direct external peerings
 neighbor EXTERNAL next-hop-self
 neighbor EXTERNAL send-community both
 neighbor EXTERNAL soft-reconfiguration inbound
 neighbor EXTERNAL prefix-list CZF in
 neighbor EXTERNAL prefix-list CZF out
 neighbor INTERNAL peer-group
 neighbor INTERNAL remote-as 64538
 neighbor INTERNAL description Internal peerings
 neighbor INTERNAL update-source dummy0
 neighbor INTERNAL next-hop-self
 neighbor INTERNAL send-community both
 neighbor INTERNAL soft-reconfiguration inbound
 neighbor INTERNAL prefix-list CZF in
 neighbor INTERNAL prefix-list CZF out
 neighbor NFX peer-group
 neighbor NFX remote-as 65532
 neighbor NFX description NFX-route-servers
 neighbor NFX next-hop-self
 neighbor NFX send-community both
 neighbor NFX soft-reconfiguration inbound
 neighbor NFX prefix-list CZF in
 neighbor NFX prefix-list CZF out
 neighbor NFX route-map NFXCZFOUT out
 neighbor NFXPUB peer-group
```

```
 neighbor NFXPUB remote-as 8251
 neighbor NFXPUB description NFX public route-servers
 neighbor NFXPUB next-hop-self
 neighbor NFXPUB send-community both
 neighbor NFXPUB soft-reconfiguration inbound
 neighbor NFXPUB prefix-list PUB out
 neighbor 10.38.15.2 peer-group INTERNAL
 neighbor 10.38.15.2 description jmnet-mh4
 neighbor 10.38.16.130 peer-group INTERNAL
 neighbor 10.38.16.130 description jmnet-ikex
 neighbor 10.38.32.242 peer-group INTERNAL
 neighbor 10.38.32.242 description jmnet-dracekk
 neighbor 10.253.32.250 peer-group NFX
 neighbor 10.253.32.251 peer-group NFX
 neighbor 81.201.48.193 peer-group NFXPUB
 neighbor 81.201.48.194 peer-group NFXPUB
 neighbor 81.201.48.195 peer-group NFXPUB
 no auto-summary
 !
access-list login remark Administrator access to zebra
access-list login permit 127.0.0.1/32
access-list login permit 10.253.32.0/24
access-list login permit 10.38.254.0/24
access-list login deny any
!
ip prefix-list CZF description CZfree.Net prefixes
ip prefix-list CZF seq 10 permit 10.0.0.0/8 ge 15 le 20
ip prefix-list CZF seq 99 deny any
ip prefix-list PUB description Public IP
ip prefix-list PUB seq 10 permit 78.108.106.0/24
ip prefix-list PUB seq 11 permit 212.79.108.0/24
ip prefix-list PUB seq 99 deny any
!
route-map NFXCZFOUT permit 10
 set metric 1000
!
line vty
 access-class login
 exec-timeout 60 0
!
```

Příloha A: konfigurace BPG routeru Sitel

```
!
! Zebra configuration saved from vty
!   2011/03/13 17:59:49
!
hostname jizak
password router
enable password tajneheslo
log file /var/log/quagga/ospf.log
!
debug ospf6 lsa unknown
!
interface eth0
 ipv6 ospf6 cost 1
 ipv6 ospf6 hello-interval 10
 ipv6 ospf6 dead-interval 40
 ipv6 ospf6 retransmit-interval 5
 ipv6 ospf6 priority 1
 ipv6 ospf6 transmit-delay 1
 ipv6 ospf6 instance-id 0
!
interface eth1
 ipv6 ospf6 cost 1
 ipv6 ospf6 hello-interval 10
 ipv6 ospf6 dead-interval 40
 ipv6 ospf6 retransmit-interval 5
 ipv6 ospf6 priority 1
 ipv6 ospf6 transmit-delay 1
 ipv6 ospf6 instance-id 0
 ipv6 ospf6 passive – OSPF neposílá do takového rozhraní oznámení
!
interface eth2
 ipv6 ospf6 cost 10
 ipv6 ospf6 hello-interval 10
 ipv6 ospf6 dead-interval 40
 ipv6 ospf6 retransmit-interval 5
 ipv6 ospf6 priority 1
 ipv6 ospf6 transmit-delay 1
 ipv6 ospf6 instance-id 0
 ipv6 ospf6 passive
!
router ospf6
 router-id 10.38.0.4
 redistribute connected
 interface eth0 area 0.0.0.0
!
line vty
!
```

Příloha B: konfigurace ospf6d routeru Jizak



Obrázek 2: Páteřní síť JM-Net o.s.
Příloha C: Páteřní síť JM-Net o.s.