Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra informačních technologií

Mechanismy přechodu mezi IPv4 a IPv6

Bakalářská práce

Autor: Martin Davídek
Studijní obor: Aplikovaná informatika

Vedoucí práce: Ing. Pavel Blažek

Hradec Králové Prosinec 2017

Prohlášení:

Prohlašuji, že jsem bakalářskou/diplomovou práci zpracoval/zpracovala samostatně a s použitím uvedené literatury.

vlastnoruční podpis

V Hradci Králové dne 19.12.2017 Martin Davídek

Poděkování:
Děkuji vedoucímu bakalářské práce Ing. Pavlu Blažkovi za metodické vedení práce, jeho hodnotné rady, doporučení a jeho cenný čas.

Anotace

Tato bakalářská práce se zkoumá internetový protokol, jeho verze a jejich možnou koexistencí. Práce je rozdělená do třech hlavních částí. První a druhá třetina jsou teoretické a poslední třetina je věnována praxi.

První část je konkrétně zaměřena na obě verze internetového protokolu (4 a 6), jejích popis, stavbu paketů, adresy a adresování. Druhá část popisuje možnosti souběžného fungování obou verzí. Zkoumá a popisuje různé možnosti resp. mechanismy, které to umožňují. Závěrečná část je věnována sérii testů, kde se pokusím dokázat, že se do IPv6 světa může připojit i běžný uživatel, bez ohledu na limity jeho připojení a že k tomu není potřeba pouze nativní IPv6 připojení.

Annotation

Title: Mechanisms of migration between IPv4 and IPv6

This bachelor thesis examines the Internet protocol, its version and their possible coexistence. The work is divided into three main parts. The first and second third are theoretical and the last third is dedicated to practice.

The first part is specifically focused on both versions of the Internet Protocol (4 and 6), its description, packet construction, addresses and addressing. The second part describes the possibilities of parallel operation of both versions. It examines and describes various options respectively mechanisms that make it possible. The final part is dedicated to a series of tests where I will try to prove that a regular user can connect to the IPv6 world, regardless of the limits of his connection, and also that native IPv6 connections are not needed.

Obsah

Seznam obrázků

Seznam tabulek

1 Úvod

V dnešní době je hlavním stavebním kamenem internetu IPv4. Počet síťových zařízení je několikanásobně větší, než je rozsah IP adres. Například v domácnosti se čtyřmi osobami může být deset takových zařízení (4x mobilní telefon, 2x notebook, 1x stolní PC, 1x tablet, 1x směrovač a 1x Smart TV). Vyčerpání adres je dlouhodobě ožehavé téma, které zatím řeší všudypřítomný překlad privátních IP adres na veřejné (NAT). Přitom od poloviny 90. let je zde nástupce a tím je IPv6, které i primárně vzniklo právě kvůli adresnímu rozsahu a snaží se vyvatovat nedostatkům svého předchůdce. IPv6 ve srovnání s IPv4 nabízí takový objem IP adres, že na jednu osobu na planetě vychází zhruba rozsah adres o velikosti celého rozsahu IPv4. Bohužel přechod od IP verze 4 k verzi 6 je běh na dlouhou trať. Je to i tím, že verze nejsou zpětně kompatibilní. Z tohoto důvodu, v průběhu let co je IPv6 na světě, vzniklo několik mechanismů. Ty umožňují koncovým uživatelům či sítím být připojen do obou světů, i když mají konektivitu jen pro jednu verzi protokolu. Zatím se jedná o připojení do IPv6 světa s IPv4 konektivitou.

Internetový protokol je službou třetí vrstvy, tj. síťové vrstvy referenčního modelu ISO/OSI. Rovněž je součástí sady Transmission Control Protocol/Internet Protocol (TCP/IP). Byl, díky jeho charakteristikám, navržen jakožto protokol s nízkou režií. Poskytuje pouze funkce nezbytné k přenosu paketu, od odesilatele k příjemci, skrze propojený systém počítačových sítí. IP také umožňuje propojovat jednotlivé lokální počítačové sítě a určuje definici způsobu směrování jednotlivých paketů mezi jednotlivými počítačovými sítěmi. (Sportack, 2004, s. 14 – 15)

Hlavními charakteristikami tohoto protokolu jsou nezávislost na typu média (Media Independent), neschopnost garance doručení paketu příjemci (Best Effort) a je nespojově orientovaný (Connectionless).

Díky tomu, že je protokolem třetí vrstvy referenčního modelu ISO/OSI, nemusí se zabývat typem média, jelikož toto mají na starost protokoly a služby nižších vrstev ISO/OSI, a tak mohou být pakety transportovány po libovolném typu média.

Sice není schopen zaručit doručení paketu příjemci, ale příčina není, že by chvíli fungoval správně a chvíli ne, nebo protože poskytuje nekvalitní komunikaci, ale je to kvůli absenci schopnosti spravovat a zotavit nedoručené nebo poškozené pakety. Pakety neobsahují žádná synchronizační data a dochází k jejich doručení v jiném pořadí, než byly odeslány. Nemají ani žádný kontrolní součet, tudíž není možné ověřit, zda nedošlo k poškození, nebo záměně dat, které paket obsahuje. Internetový protokol ani nemá možnost oznámit odesilateli, že se nezdařilo odeslání paketu nebo došlo k jeho poškození. Za řešení všech těchto úskalí mají zodpovědnost protokoly a služby čtvrté (transportní) vrstvy referenčního modelu ISO/OSI, jako je například protokol Transmission Control Protocol (TCP). Pakety obsahují pouze informace o tom, odkud byl paket odeslán a kam má být odeslán.

Síťová vrstva včetně internetového protokolu, se nestará o transport paketů mezi odesilatelem a příjemcem nebo mezi jednotlivými uzly v počítačové síti. Nemá na starosti a povědomí o typu komunikace, který je obsažen v paketu. IP je nespojově orientovaný a to znamená, že mezi odesilatelem a adresátem není před odesláním paketu vytvořeno žádné přímé spojení (end-to-end). (Parziale et al., 2006, s. 68)

Celé to můžeme přirovnat k odesílání dopisu poštou. Odesilatel podá dopis na poště. Pošta bude dopravovat dopis různými prostředky, jako např. dodávka a vlak. Dále nemá žádný přehled o obsahu dopisu, pouze na obalu je adresa, která ale neposkytuje informaci o tom, je-li adresát dostupný, zda bude dopis příjemci doručen nebo jestli si ho přečte a navíc příjemce neví dopředu, že mu má být doručen nějaký dopis.

Tolik k úvodu do internetového protokolu, níže jsou jeho dvě verze. Tato práce má další dvě části. Druhá část se budeme zaobírat úvodem do problematiky přechodu mezi IPv4 a IPv6 a mechanismy, které umožňují koexistenci obou protokolů. Zároveň se bude věnovat tunelování. Ve třetí, tedy poslední části práce, je otestován v praxi jeden z mechanismů.

2 Zapouzdření

Jednou z funkcionalit internetového protokolu je zapouzdření (encapsulation), ke které dochází při odesílání paketu. Při zapouzdření na třetí (síťové) vrstvě referenčního modelu ISO/OSI se převezme segment, který je vytvořen při zapouzdření na čtvrté (transportní) vrstvě, připojí se k němu IP hlavička a tím vzniká paket. Ten je následně předán druhé (datové) vrstvě ISO/OSI k přípravě na odeslání. Hlavička paketu obsahuje informace, mezi které patří i IP adresa odesilatele a IP adresa příjemce, sloužící k doručení paketu adresátovi.

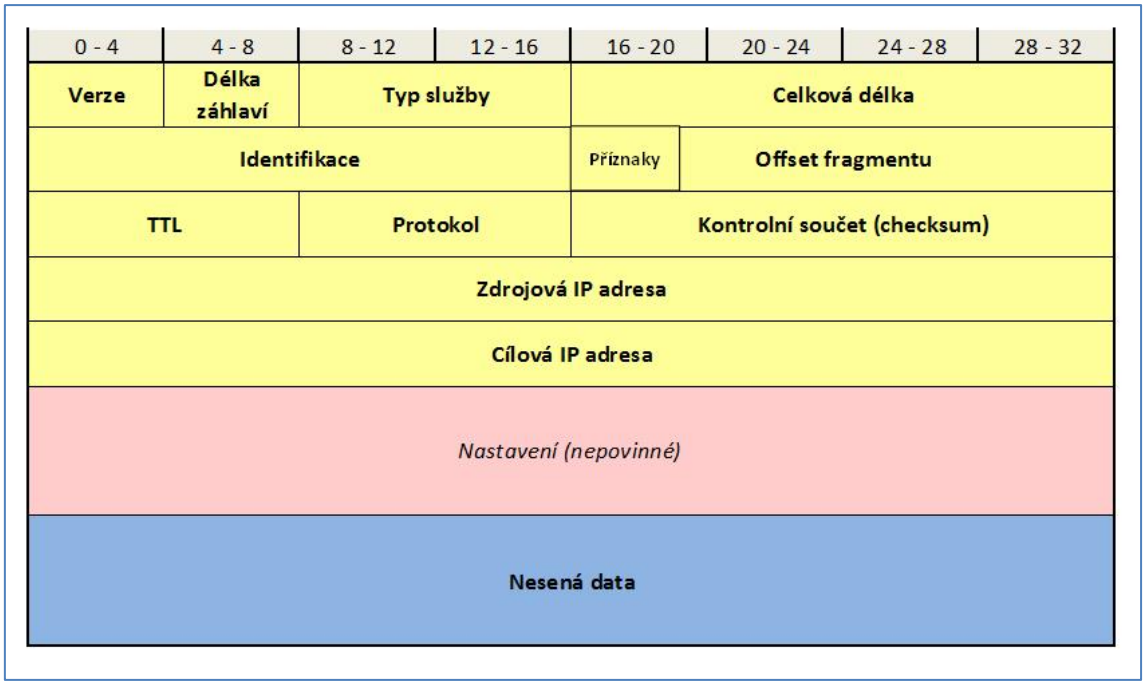
V opačném případě mluvíme o tzv. vypouzdření (decapsulation). Při vypouzdření síťová vrstva obdrží od datové vrstvy paket a provede se kontrola, zda je cílová IP adresa obsažená v hlavičce paketu shodná s IP adresou zařízení. Pokud se adresy shodují, dojde k odebrání IP hlavičky a zbylá část (segment) se předá transportní vrstvě k dalšímu zpracování, jinak je paket směruje k dalšímu zařízení nebo dojde k zahození paketu. (Learn Networking, 2007)

3 Internetový protokol verze 4

IPv4 je v dnešní době mnohem více využívaná v počítačových sítích globálně po celém světě než novější verze. Jak už bylo řečeno v úvodu, patří do sady Transmission Control Protocol/Internet Protocol (TCP/IP) a pracuje na třetí (síťové vrstvě) referenčního modelu ISO/OSI. Byl popsán roku 1981 organizací Internet Engineering Task Force (IETF) v dokumentu RFC 791: Internet Protocol. Jeho původní nasazení bylo pro Ministerstvo Obrany Spojených států amerických (DoD – Department of Defence), jako náhrada původní vojenské sítě ARPANET. Používá adresování s využitím 32 bitové IP adresy. V počítačové síti používající internetový protokol verze 4 má každý uzel a koncové zařízení nastavené svou IP adresu a masku sítě. IP adresy, masky a adresování si popíšeme později. Dále umožňuje fragmentaci a defragmentaci paketů, které je třeba rozdělit na menší části, pokud je to vyžadováno kvůli nižší šířce pásma v počítačové síti a v IP hlavičce jsou k tomu vyhrazena pole *identifikace*, *příznaky* a *offset fragmentu*. (RFC 791: Internet Protocol, 1981)

3.1 IPv4 paket

Paket je protokolová datová jednotka (PDU - Protocol Data Unit). Protokolová datová jednotka je složena z informací dané vrstvy referenčního modelu ISO/OSI a protokolové datové jednotky vyšší vrstvy (Institute for Telecommunication Sciences, 2007). Každá vrstva má svojí jednotku a paket je protokolovou datovou jednotkou třetí (síťové) vrstvy. Jak už bylo zmíněno, paket je složen z IP hlavičky a segmentu (protokolová datová jednotka transportní vrstvy ISO/OSI), jak nám demonstruje Obrázek 1. Nejdůležitější částí paketu je IP hlavička, jenž sestává z několika polí, která jsou popsána níže.



Obrázek 1: Formát IPv4 datagramu [9]

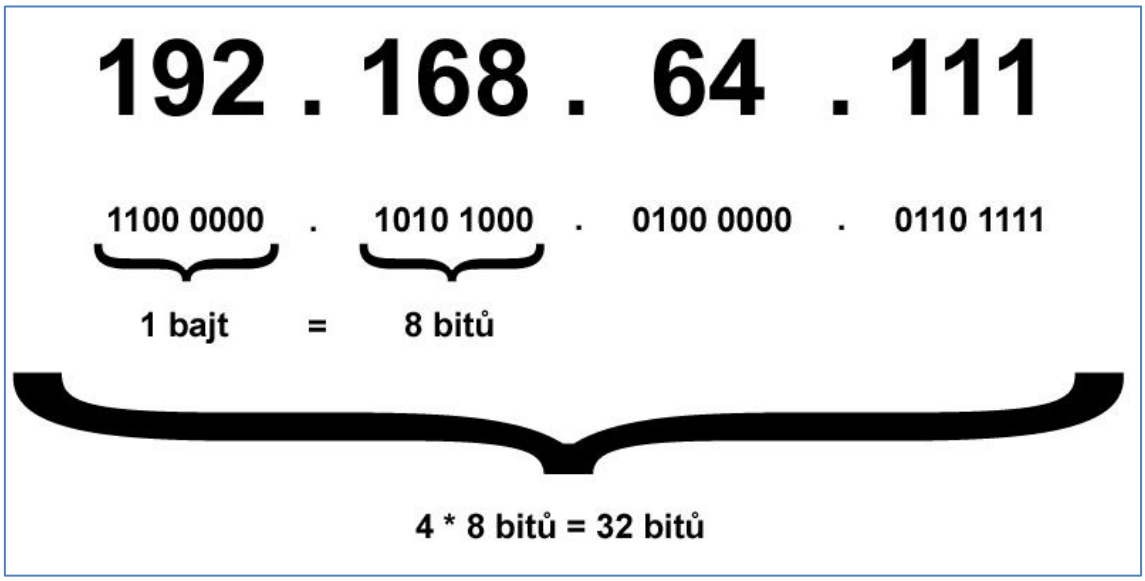
Pole IP hlavičky:

Zdroj: RFC 791: Internet Protocol, s. 11 – 14

- Verze: udává verzi protokolu (pro IPv4 je binární hodnota 0100)
- Délka záhlaví: udává velikost IP hlavičky (výchozí hodnota je 20B a maximální 60B)
- Typ služby: určuje prioritu každého paketu
- Celková délka: udává velikost paketu i s aplikačními daty (maximálně 65535 B)
- Identifikace: unikátní identifikátor původního paketu
- Příznaky: určuje, zda je paket fragmentován a jak
- Offset fragmentu: určuje polohu fragmentu v původním paketu (je-li binární hodnota nulová, paket není fragmentován)
- TTL (Time To Live): určuje životnost paketu; hodnota je při každém průchodu směrovačem dekrementována (pokud je hodnota nulová, pak je paket zahozen); zabraňuje zacyklení paketu v počítačové síti
- Protokol: určuje protokol transportní vrstvy ISO/OSI, kterému je paket předán (například protokol TCP je vyjádřen hodnotou 6)
- Kontrolní součet (checksum): kontrolní hodnota; po doručení paketu se provede kontrolní součet a hodnota se porovná s kontrolní hodnotou (při neshodě hodnot je paket zahozen)
- Zdrojová adresa: zdrojová IP adresa
- Cílová adresa: cílová IP adresa

3.2 IPv4 adresy

IPv4 používá IP adresy o velikosti 4 bajty respektive 32 bitů. Je to binární hodnota reprezentovaná v dekadické notaci a je rozdělena tečkami na čtyři části neboli oktety. Příklad zápisu pomocí binární a dekadické notace můžeme vidět na Obrázku 2. Každý oktet obsahuje 8 bitů a nabývá hodnot od 0 (binárně 00000000) do 255 (binárně 11111111), z toho vyplývá, že adresní prostor internetového protokolu verze 4 má rozsah 0.0.0.0 až 255.255.255.255 tzn. 2^{32} IP adres. Dostálek (2000, s. 155) uvádí, že IP adresa rozdělena na dvě části a to na část označující počítačovou síť a část označující síťové zařízení v počítačové síti, toto rozdělení je podrobněji popsáno v následující podkapitole. (Parziale et al., 2006, s. 68 – 69)



Obrázek 2: Struktura IP adresy

Existuje organizace Internet Assigned Numbers Authority (IANA), která se mimo jiné stará o správu a koordinaci rozsahu IP adres a redistribuci částí tohoto rozsahu (bloků IP adres) mezi regionální registry (RIR - Regional Internet Register), které také spravují IP adresy a redistribuují adresy jednotlivé poskytovatele internetu (ISP – Internet Service Provider) v daném regionu. Na světě je pět regionálních registrů. (Kozierok, 2005a, s. 55 – 57)

Přehled regionálních registrů:

Zdroj: Kozierok 2005a s. 56 – 57

- Evropa, střední a středovýchodní Asie - RIPE NCC (Reseaux IP Europeans)
- Severní Amerika - ARIN (American Registry for Internet Numbers)
- Jižní Amerika a některé Karibské ostrovy - LACNIC (Regional Latin-American and Caribbean IP Address Registry)
- Asie kromě střední a středovýchodní a Pacifik - APNIC (Asia Pacific Network Information Centre)
- Afrika - AfriNIC (African Network Information Centre)

3.3 Adresování v IPv4 sítích

Z předchozí kapitoly víme, že se IP adresa skládá z části určující počítačovou síť a části, která přímo určuje konkrétní síťové zařízení. Není možné zjistit pouze z adresy sítě, do jaké počítačové sítě zařízení patří, kde končí síťová část a kde část pro síťové zařízení začíná. Každé zařízení ve směrované počítačové síti je identifikováno nejen IP adresou, ale také musí mít nastavenu masku sítě, aby bylo možné přesně určit, kam zařízení patří. Všechna v téže síti, musí mít stejnou masku a IP adresu z rozsahu počítačové sítě, do které patří.

Maska určuje, jaký objem zařízení může být v jedné síti. Výpočet je prováděn tak, že se maska sítě převede do binární soustavy, následuje inverze této binární hodnoty a výslednou binární hodnotu převedeme do dekadické soustavy, k hodnotě přičteme hodnotu jedna a výsledná hodnota je počtem adres sítě. Příklad 1 toto názorně demonstruje. Pro úplnost jsou odečteny dvě IP adresy od výsledného počtu. Ty jsou vyhrazené pro adresu sítě a broadcast. Obě se nesmí použít pro identifikaci síťového zařízení. Adresa sítě je první z rozsahu a je identifikátorem sítě. Broadcast je poslední adresa z rozsahu a slouží pro komunikaci se všemi síťovými zařízeními v síti. Výše zmíněné typy IP adres jsou zobrazeny na Obrázku 3.

Příklad 1: výpočet počtu adres v IPv4 síti

maska sítě 255.255.255.192

1. *Masku převedeme na binární tvar: 11111111 11111111 11111111 11000000*
2. *Invertujeme binární hodnotu masky: 00000000 00000000 00000000 00111111*
3. *Převedeme na dekadický tvar a doplníme: 111111B = 63 + 1 = 64 IP adres*
4. *Od výsledku odečteme hodnotu 2: 64 - 2 = 62 IP adres použitelných pro adresování*

Další využití masky sítě je k výpočtu síťové adresy. Když máme IP adresu (například počítače) s maskou sítě, výpočet IP adresy sítě slouží pro zjištění adresy sítě, do které IP adresa patří. Provede se logický součin binárních hodnot jednotlivých oktetů IP adresy a masky sítě. Následně už jen stačí jednotlivé oktety převést zpět na zápis v dekadickém tvaru, viz Obrázek 3. (Kozierok, 2005a, s. 247 – 250)

IP adresa:	192.168.64.111				
Maska síte:	255.255.0.0				
Binární součin (AND)	11000000	10101000	01000000	01101111	IP adresa (bin.)
	11111111	11111111	00000000	00000000	Maska sítě (bin.)
	11000000	10101000	00000000	00000000	IP adresa sítě (bin.)
IP adresa sítě:	192.168.0.0				
Broadcast:	192.168.255.255				
Rozsah IP adres:	192.168.0.1 až 192.168.255.254				

Pozn:	
	1 AND 1 = 1
	1 AND 0 = 0
	0 AND 0 = 0

Obrázek 3: Výpočet sítě

4 Internetový protokol verze 6

Internetový protokol verze 6, někdy taky označovaný jako internetový protokol další generace (IPng - IP Next Generation), je nástupcem internetového protokolu verze 4. Na začátku 90. let 20. století se začali objevovat obavy z problémů spojených s internetovým protokolem verze 4. Na základě těchto obav začala organizace Internet Engineering Task Force (IETF) hledat nové řešení. Tím byl vývoj nové verze internetového protokolu. V roce 1995 tak vzniká nová verze IP a vydání dokumentu RFC 1883: Internet Protocol, Version 6 (IPv6) Specification popisující tento protokol. Dokument RFC 1883 byl v roce 1998 nahrazen upraveným dokumentem RFC 2460: Internet Protocol, Version 6 (IPv6) Specification, který obsahuje aktuální popis internetového protokolu verze 6.

Přináší spoustu nových vylepšení. Nejvýznamnějším rozšířením je extrémní navýšení rozsahu IP adres, které internetový protokol verze 6 nabízí, z čehož plyne, že v IPv6 počítačové síti je možné přímo adresovat mnohem více zařízení než v IPv4 síti. S rozšířením adresného prostoru vzniklo i více úrovní v hierarchii IP adres. Byla také zjednodušena i jejich automatická konfigurace.

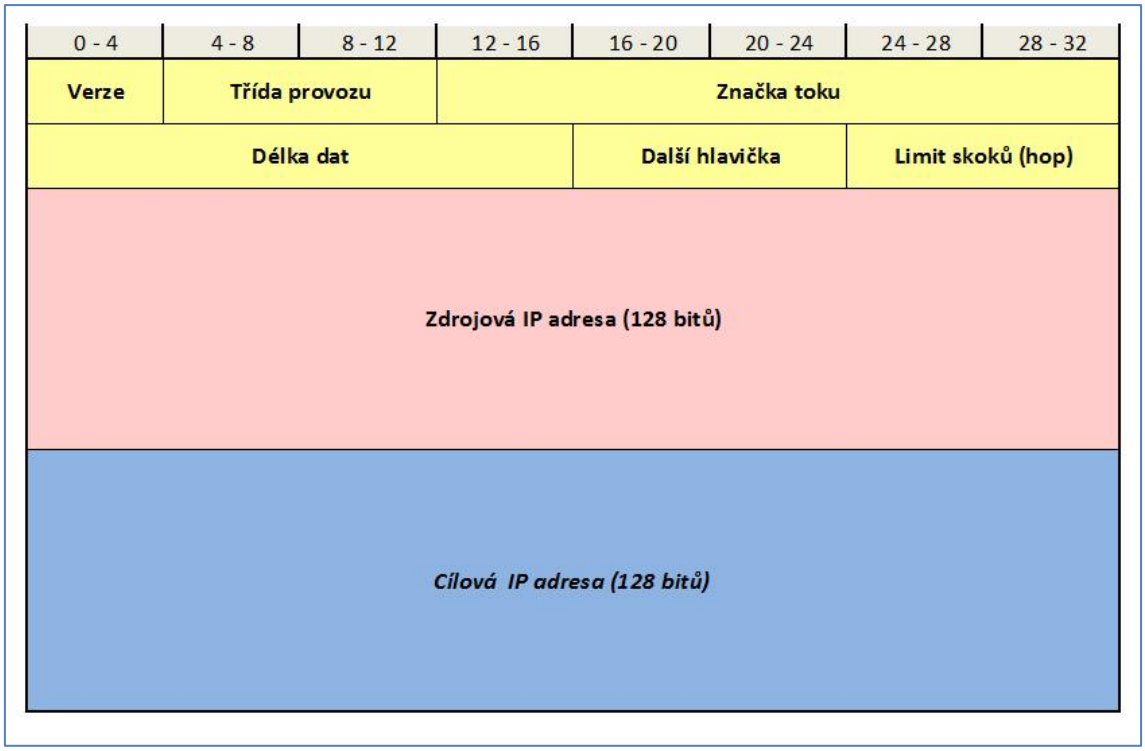
Inovaci prodělaly i IPv6 pakety. IP hlavička paketu je značně zjednodušena snížením počtu polí, které hlavička obsahuje. Většina polí hlavičky IPv4 paketu byla odstraněna a nahrazena. Zjednodušení značně ulehčuje manipulaci s ní, obzvlášť při směrování. Dále došlo ke změnám v kódování hodnot IPv6 hlavičky, a to umožňuje směrovacím zařízením efektivnější směrování. Limitace velikosti jednotlivých polí IP hlavičky jsou u IPv6 paketu nižší. Je rovněž mnohem flexibilnější vůči zavádění budoucích rozšíření. Byla přidána i funkce označování IPv6 paketů patřící do speciálních komunikačních kanálů. Detailněji ho popisuje podkapitola, která je tomu věnována.

Dalším významným přínosem internetového protokolu verze 6 je, že odpadá nutnost využívat mechanismus NAT (Native Address Translation) pro překlad IP adres, protože adresný prostor obsahuje enormní množství IP adres, což umožňuje přidělit každému zařízení v internetu jeho vlastní IP adresu a tím odpadá problém s přímou konektivitou mezi koncovými zařízeními (end-to-end). Internetový protokol verze 6 mimo jiné implementuje i integrované zabezpečení. Nativně jsou podporovány schopnosti autentizace, integrity dat a ochrany osobních údajů. (Deering et al. 1998)

Jako zajímavost bych rád uvedl, že bylo vydáno usnesení Vlády ČR č. 727 o přechodu na internetový protokol verze 6, ve kterém Vláda ČR souhlasí s přechodem na tento protokol a zároveň ukládá ministrům a vedoucím pracovníkům úředních orgánů státní správy zajistit „*od 30. června 2009 při pravidelné obnově síťových prvků jejich kompatibilitu s internetovým protokolem verze 6 (IPv6)*“ a „*do 31. prosince 2010 přístup k internetovým stránkám a veřejně dostupným službám eGovernmentu internetovým protokolem verze 4 (IPv4) i internetovým protokolem verze 6 (IPv6)*“ (Usnesení Vlady ČR č. 727 2009).

4.1 IPv6 paket

Jak už bylo zmíněno v podkapitole o IPv4 paketu, paket je protokolová datová jednotka (PDU - Protocol Data Unit) třetí (síťové) vrstvy referenčního modelu ISO/OSI a skládá se z IP hlavičky a segmentu, neboli PDU druhé (datové) vrstvy modelu ISO/OSI. Hlavička IPv6 paketu byla razantně upravena ve srovnání s IPv4 hlavičkou. Satrapa (2012, s. 35) uvádí, že velikost IPv6 paketu je oproti starší verzi dvojnásobná tzn., že velikost byla navýšena na 40 bajtů. Změnu prodělala i struktura IP hlavičky. Místo 12 polí, které obsahuje hlavička IPv4 paketu, byl počet polí zredukován na osm. V hlavičce IPv6 paketu byla ponechána pouhá tři pole, která jsou obsažena i v hlavičce IPv4 paketu. Jsou to pole pro verzi internetového protokolu, zdrojovou a cílovou IP adresu. Ostatní pole byla odstraněna a nahrazena pěti novými poli viz Obrázek 4. V rámci úpravy IPv6 paketu byla přidána již zmíněná funkce označován, které patří do speciálních komunikačních kanálů na základě požadavku odesilatele na zvláštní manipulaci s IPv6 pakety, jako nadstandardní real-time službu nebo Qualtity of Services (QoS), jak uvádí Deering et al. (1998).



Obrázek 4: Formát IPv6 hlavičky [12]

Pole IP hlavičky:

Zdroj: Deering et al. 1998

- Verze: udává verzi protokolu (pro IPv6 je binární hodnota 0110)
- Třída provozu: určuje zařazení paketu do určité přepravní třídy (Traffic Class), nebo vyjadřuje prioritu paketu
- Značka toku: identifikátor paketů se stejnými vlastnostmi
- Délka dat: udává velikost paketu bez standardní hlavičky, ale velikosti rozšiřujících hlaviček se započítávají
- Další hlavička: udává typ dat, které paket obsahuje, anebo udává druh rozšiřující hlavičky
- Limit skoků (hop): ekvivalent k poli TTL, které se používá v hlavičce IPv4 paketu; princip zůstává stejný, tzn. hodnota je dekrementována při každém průchodu směrovacím zařízením, je-li hodnota nulová, je paket zahozen, a tak stejně jako TTL slouží k eliminaci zacyklení paketu v počítačové síti
- Zdrojová adresa: zdrojová IP adresa
- Cílová adresa: cílová IP adresa

4.2 IPv6 adresy

Velikost IPv6 adresy byla zvětšena na 128 bitů a celkový rozsah tím vzrostl na 2^{128} IPv6 adres. Oproti tomu internetový protokol verze 4 poskytuje rozsah obsahující pouze 2^{32} IP adres o velikosti 32 bitů. IPv6 adresy jsou taktéž binární hodnoty, ale proti IPv4 adresám se používá hexadecimální notace pro jejich zápis. Jsou rozděleny na osm částí, které jsou odděleny dvojtečkami. Každá část neboli hextet má velikost 16 bitů a může nabývat hodnot od 0000_H (binárně 0000 0000 0000 0000) do $ffff_H$ (binárně 1111 1111 1111 1111), z čehož vyplývá, že rozsah IP adres je od 0000:0000:0000:0000:0000:0000:0000:0000 do ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff.

Poněvadž je zápis IPv6 adresy dlouhý, je možné ho zkracovat. Má-li celý hextet nulovou hodnotu (0000_H), dá se zapsat jen pomocí jedné nuly. Dále, pokud je hextet nenulový a obsahuje na začátku jednu nebo více nul, nuly se odstraní. V případě že nezačíná nulou, není možné zápis takového hextetu zkrátit. Při zkracování zápisu IP adresy je možné nahradit její část, která obsahuje nejvíce nul, dvěma dvojtečkami. Musí se však dodržet pravidlo, kdy je přípustné odstranit nuly, které jsou na začátku hextetu, tj. za dvojtečkou vymezující jeho začátek. Pro ukázku nám poslouží Příklad 2. (Satrapa, 2012, s. 56 – 57)

Příklad 2: zkracování IPv6 IP adres

IP adresa: 2001:0db8:0000:0000:0fff:0000:00e3:10c0

1. *Nahradíme část s nejvíce nulami dvěma dvojtečkami:*
   2001:0db8::fff:0000:00e3:10c0
2. *Odstraníme počáteční nuly z hextetů, u kterých je to možné:*
   2001:db8::fff:0:e3:10c0

4.3 Adresování v IPv6 sítích

Všechna síťová zařízení v IPv6 sítích patřící do stejné sítě, musí mít rovněž IP adresu ze stejného rozsahu dané počítačové sítě. Změnou, oproti adresování v IPv4 sítích, je absence masky sítě. IPv6 používá prefix sítě, který je zapisován za adresou a odděluje se lomítkem, např. 2001:db8::/64. Určuje rozsah (v bajtech), který je možný použít k adresování v síti. Internetový protokol verze 6 umožňuje, aby zařízení mělo přiřazeno více než jednu IPv6 adresu na každém síťovém rozhraní. Navíc jsou definovány IP adresy, které musí mít zařízení přiřazené (pro počítač např. lokální linková adresa pro každé rozhraní, přidělená skupinová, atd.) a je schopné využívat všechny tyto adresy pro komunikaci, pokud je to nutné.

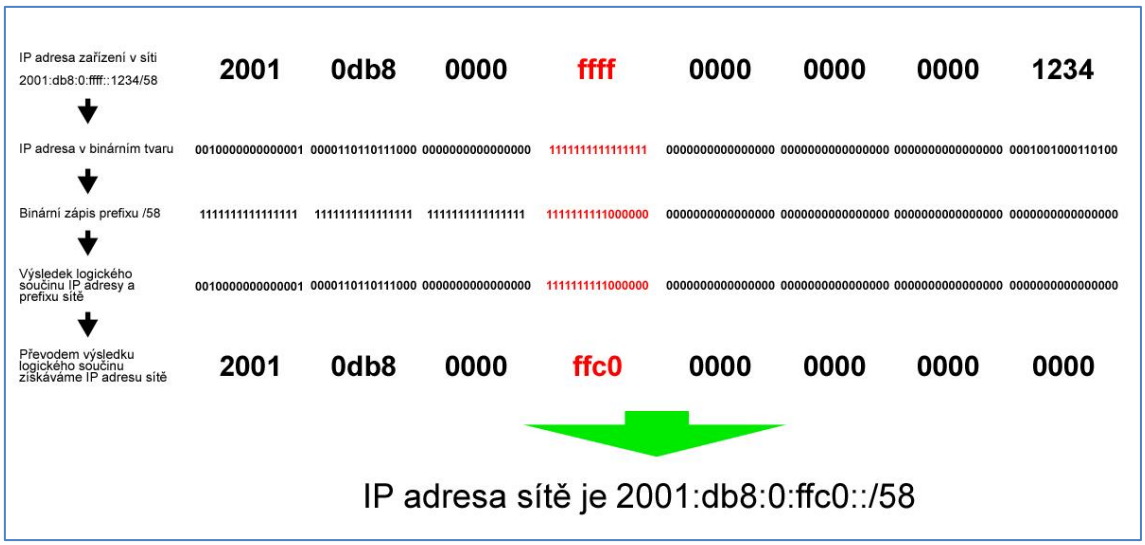
Se zavedením síťového prefixu, namísto masky sítě, je možno jednoduše zjistit rozsah sítě. V první řadě se odečte od rozsahu, který IPv6 nabízí, hodnota obsažená v prefixu a následně se umocní výslednou hodnotou číslo 2. Po umocnění je získán výsledný počet IP adres dané sítě. Je-li potřeba pro úplnost znát počet adres použitelných pro síťová zařízení, odečteme od výsledného počtu hodnotu 2, viz Příklad 3. První IP adresa je, obdobně jakou IPv4 adresování, rezervována pro adresu sítě a poslední je rezervována pro anycast adresu (nahrazuje broadcast adresu).

Příklad 3: výpočet počtu adres v IPv6 síti:

Prefix: /110

1. *Od rozsahu internetového protokolu verze 6 odečteme hodnotu prefixu:*
   128 – 110 = 18
2. *Výsledným číslem umocníme číslo 2:*
   218 = 262144 IP adres
3. *Od výsledku odečteme hodnotu 2:*
   262144 – 2 = 262142 IP adres použitelných pro adresování

Prefix sítě se využívá i při výpočtu IP adresy sítě. Pro zjištění, do jaké sítě patří daná IP adresa, je nutné znát i prefix sítě. Potom je možné výpočet IP adresy podsítě provést. Pro výpočet se provádí, podobně jako u IPv4 adresování, logický součin (AND) IP adresy a prefixu sítě. Prefix určuje kolik počátečních bitů z celkových 128, má hodnotu 1. Z předchozí věty plyne, že prefix převedený na binární číslo má 128 číslic a prvních *n* číslic jsou jedničky, kde *n* je rovno hodnotě prefixu, a zbylé jsou nuly. Po provedení logického součinu je binární číslo převedeno zpět na hexadecimální tvar čímž získáme IPv6 adresu sítě, jak můžeme vidět na obrázku 5. (Kozierok 2005a s. 373 – 381, Satrapa 2012 s. 58)



Obrázek 5: Výpočet IP adresy sítě v IPv6

4.4 Výhody oproti předchozí verzi

Zřejmě se v této podkapitole zopakují již zmíněné informace, ale je důležité shrnout všechny výhody. Nepochybně na prvním místě je rozsah adres, který je ohromný. Ve srovnání je 2^{32} IPv4 adres oproti 2^{128} IPv6 adres, což řeší problém s nedostatkem IP adres. Úpravou prošla i hlavička paketu, která se zjednodušila za účelem rychlého zpracování. Sice došlo u nové verze k navýšení hlavičky z 20 bajtů na 40 bajtů, ale z toho je 32 bajtů vyhrazeno pro IP adresy příjemce a odesilatele. V porovnání s předchozí verzí byly, díky zjednodušení, uspořeny 4 bajty, jelikož u IPv4 zabírá IP adresa příjemce a odesilatele 8 z 20 bajtů. Zároveň je i samotný IPv6 paket větší a to nejen příčinou vetší hlavičky, ale i kvůli většímu prostoru pro data. Další předností je automatická konfigurace pomocí bezstavové konfigurace a DHCPv6, která

umožňuje nově připojeným zařízením získat od implicitního směrovače informaci pro konfiguraci síťového rozhraní. Ustoupilo se také od kontrolního součtu (checksum), který u IPv4 sloužil ke kontrole, zda paket dorazil celý, nebo došlo cestou k jeho poškození. S nasazením IPv6 i odpadá potřeba používat překlad adres (NAT) díky adresnému prostoru. Nová verze tedy evidentně nabízí jisté výhody oproti staré a v další kapitole si povíme, jak jí můžeme jít naproti, i když jsme omezeni IPv4 konektivitou. (Satrapa, 2012, s. 17 – 287)

5 Mechanismy

V dnešní době je internet složen hlavně z nativních IPv4 sítí, dále pak z nativních IPv6 sítí a duálních sítí, které fungují, jak na internetovém protokolu verze 4, tak i na verzi 6. Vzhledem k rozšířenosti IPv4 v internetu, a vůbec dnešnímu globálnímu využití počítačových sítí, je přechod na IPv6 běh na dlouhou trať. Dá se předpokládat, že i po globálním přechodu na IPv6 se i nadále budou vyskytovat sítě, které budou využívat stále IPv4 a uživatelé nebudou chtít být nikterak omezováni konektivitou, do sítí využívající pouze jednu z verzí internetového protokolu, a proto je zapotřebí mechanismů, které zajistí konektivitu do všech sítí bez ohledu na používanou verzi internetového protokolu.

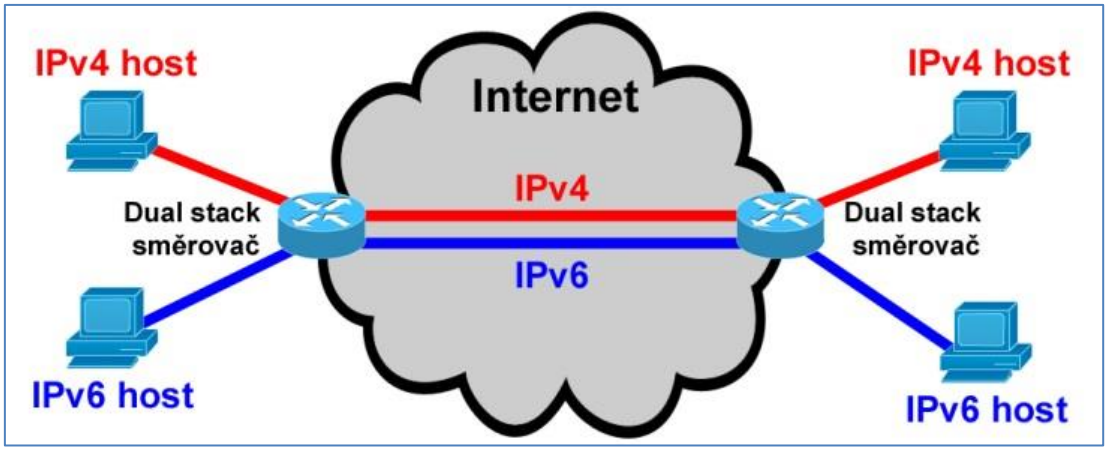
V této kapitole jsou popsány jednotlivé mechanismy a jejich rozdělení do jistých skupin. Tyto mechanismy, kterým jsou věnovány následující podkapitoly, umožňují to, že obě verze internetového protokolu mohou společně koexistovat v počítačových sítích a vzájemně spolupracovat. Kromě koexistence a spolupráce obou protokolů navrhla organizace Internet Engineering Task Force (IETF) tyto mechanismy za účelem umožnit bezproblémový a plynulý přechod z internetového protokolu verze 4 na jeho nástupce internetový protokol verze 6.

Během řady uplynulých let vývoje IPv6 vznikla řada přechodových mechanismů, některé z nich skončily neúspěchem a byly zavrženy, ale určitě můžeme očekávat, že budou vznikat další. Využití těchto nástrojů má však, kromě výhod, i svá úskalí. Při nasazení je potřeba většinou použít síťové prvky s výkonnějším procesorem. Další problém je s vyšší latencí, zejména u tunelování, a s vyšší ztrátovostí paketů.

Jak již bylo zmíněno, vznikla řada mechanismů a ty jsou popsány v následující část práce, tedy jen některé z těch známějších. Dají se rozdělit na tři kategorie podle toho, jaký je princip jejich fungování. Dělí se na tzv. dvojí zásobník (podle anglického názvu dual stack), tunelovací mechanismy a překladače. (Nordmark et al. 2005)

5.1 Dvojí zásobník

Dvojí zásobník, spíše známý pod anglickým názvem dual stack, je ve své podstatě nejprimitivnější a zároveň nepříliš praktické řešení, jak využívat oba internetové protokoly zároveň. Jde se o zařízení, která jsou nakonfigurovaná pro obě verze internetového protokolu a na jedné fyzické síti jsou tedy dvě logické sítě, viz Obrázek 6. Takové zařízení pak rozhodne, kterou verzi použije pro odeslání dat s tím, že k rozhodování dochází na aplikační vrstvě. Toto řešení s sebou nese i finanční náklady navíc, například pro koncového uživatele, který musí platit poskytovateli internetu za konektivitu na obou protokolech nebo náklady na zařízení, jako například směrovač který bude muset zvládat větší provoz a to na obou protokolech. Tomuto mechanismu není potřeba dále věnovat pozornost. Ta bude věnována tunelování a jednotlivým tunelovacím mechanismům. (Filip 2011, Satrapa 2003)

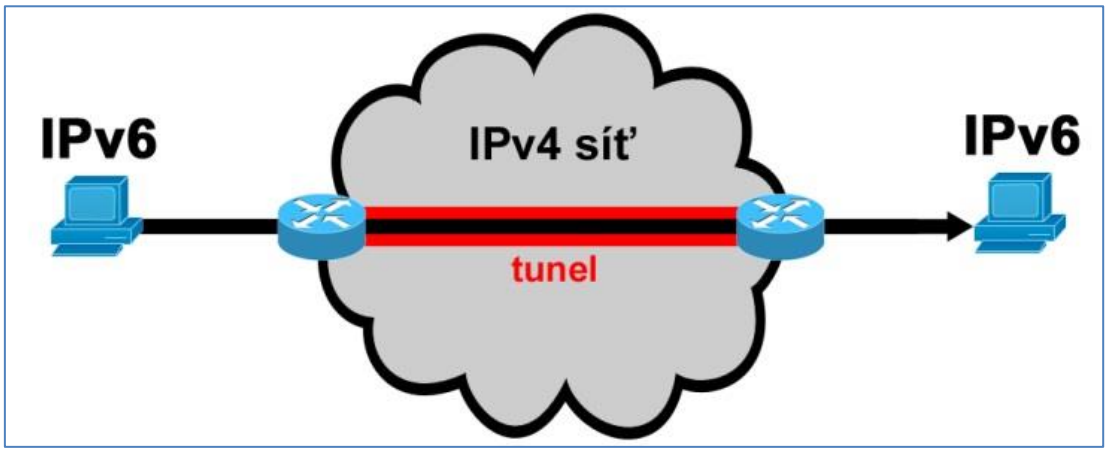


Obrázek 6: Dual Stack

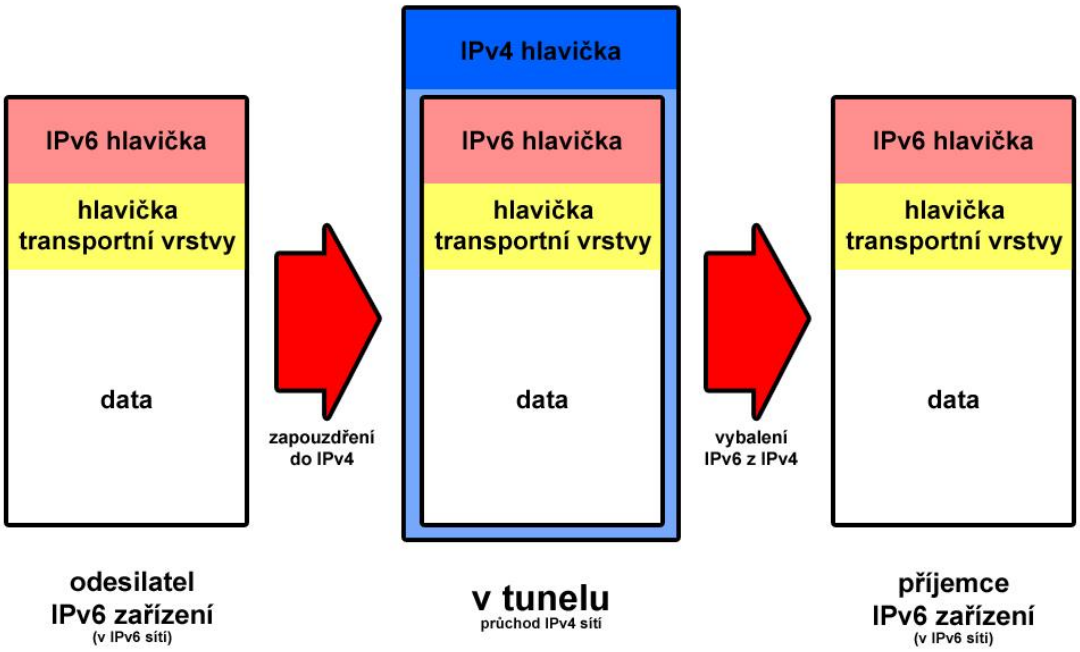
5.2 Tunelování

Na rozdíl od dvojího zásobníku, kdy je konektivita pro oba internetové protokoly a zařízením může podle potřeby využít IPv4 nebo IPv6, mechanismus tunelování je právě určen pro situaci, kdy mezi dvěma zařízeními, která fungují na stejné verzi internetového protokolu, neexistuje přímé spojení a data musí urazit minimálně část své cesty sítí fungující na druhé verzi internetového protokolu. V dnešní době nepochybně značně se jedná tunelování IPv6 skrze IPv4 síť, a proto se tato kapitola převážně věnuje této kategorii mechanismů. Samozřejmě je možná i opačná varianta, ale o tom až později.

Pro demonstraci je uvažována komunikace mezi dvěma IPv6 zařízeními, jejíž cesta je z části tvořena IPv4 sítí, viz Obrázek 7 a daný příklad nastíní princip tunelování. Odesilatel odešle IPv6 datagram a ten cestuje IPv6 sítí, dokud nenarazí na hraniční směrovač mezi IPv6 a IPv4 síti, který musí být nakonfigurovaný pro obě verze internetového protokolu. Tento směrovač pak IPv6 datagram zabalí do IPv4 datagramu, u kterého v hlavičce nastaví cílovou IP adresu na IP adresu hraničního směrovače s IPv6 sítí, skrze kterou bude IPv6 datagram dále pokračovat. Jakmile IPv4 datagram dorazí na hraniční směrovač, směrovač z něj vybalí původní IPv6 datagram s cílovou IP adresou příjemce a ten odešle po IPv6 síti směrem k příjemci. Průběh datagramu nám znázorňuje Obrázek 8. Z pohledu IPv6 datagramu je celá cesta IPv4 sítí považovaná za jeden skok (anglicky hop). (Sportack 2004 s. 318 – 319, Conta et al. 1998)



Obrázek 7: IPv6 tunel



Obrázek 8: IPv6 datagram - průchod tunelem

Tunelování s sebou nese i jisté nevýhody. Vyplývají právě z onoho principu zapouzdření jednoho protokolu do druhého protokolu. Původní datagram plní roli nesených dat v datagramu, který ho zapouzdřuje a je k němu přidána hlavička a patička protokolu. To může, pokud by byla omezená propustnost linky, vést k fragmentaci a problémům s firewallem, jelikož transportní hlavička je obsažena v prvním fragmentu. Potíže s firewallem nastanou hlavně, když bude striktně nastavený. Tunelované datagramy budou blokovány, protože mají nastavenou v hlavičce datagramu, konkrétně v poli protokol, hodnotu 41. Již zmíněná fragmentace a také skutečnost, že cesta tunelem, zejména statickým, nemusí být ta nejkratší a nejvhodnější. Z hlediska topologie může mít za následek vyšší latenci. Zároveň také vzniká větší zátěž na výkon hraničních zařízení, která musí datagramy rozbalovat nebo zabalovat popř. fragmentovat. (Hlaváček 2011 s. 60)

Jednotlivé tunelovací mechanismy se mohou ještě rozdělit na dvě skupiny, přesněji na konfigurované a automatické. Konfigurované tunelování spočívá v tom, že se nakonfigurují oba konce tunelu a vytvoří se spojení. Tato variantu je použitelná pro permanentní připojení zařízení nebo sítě skrz tunel. Pro snadnější zprostředkování tunelů vznikla mechanika tunel server, později pak tunel server/broker, která částečně automatizuje proces vytvoření tunelu, ale pořád se jedná o konfigurované tunelování. Tématem tunel server/broker se věnuje hned následující podkapitola. Automatické tunelování se od konfigurovaného liší tím, že tunel je vytvořen automaticky některým z tunelovacích protokolů jako 6to4, Teredo aj. a tyto podkapitoly následují po tunel server/broker. (Steffann et al. 2013)

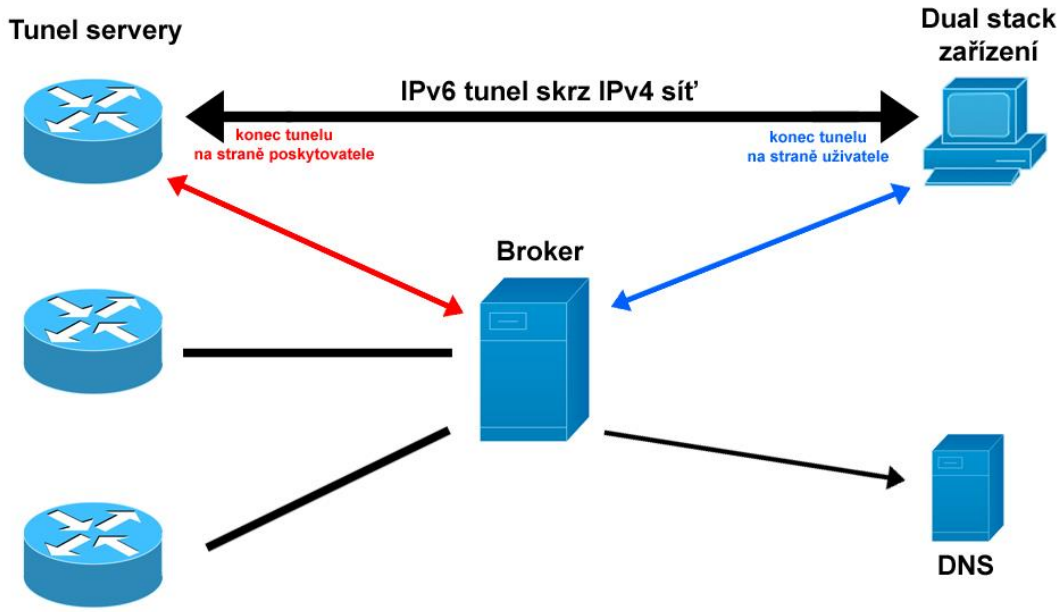
5.2.1 Tunel server/broker

Jak už název napovídá, tento tunelovací mechanismus se skládá ze dvou částí, respektive dvou zařízení, i když na začátku se jednalo jen o jedno zařízení. Jednalo se pouze o tunel server. Ten je opačným koncem IPv6 tunelu od uživatele či hraničního směrovače IPv6 sítě bez přímé konektivity do IPv6 internetu. Jak už vyplývá z předchozí věty, tunel server je zařízení s IPv4 a IPv6 konektivitou a stará se o chod jednoho či více tunelů. Pro získání tunelu se uživatel většinou musí registrovat přes web. Poskytovatel tunel serveru na základě informací od uživatele nakonfiguruje tunel a uživateli zašle konfigurační skript. Uživatel na svém zařízení provede konfiguraci pro připojení k tunelu.

„*Jelikož se v praxi ukázalo, že rozhraní pro uživatele a vlastní realizace tunelu jsou dva dost odlišné úkoly, došlo k rozdělení práce.*“ (Satrapa, 2003) Tunelovací mechanismus tunel server byl rozšířen o tunel broker. Provoz a řízení přidělených tunelů zůstává stále v režii tunel serveru. Server je typicky směrovač, který si poradí s větší zátěží, dokáže provozovat několik tunelů najednou a musí mít konektivitu jak do IPv4 tak i do IPv6.

Broker lze považovat za virtuálního ISP poskytujícího IPv6 připojení uživatelům, jenž mají k dispozici pouze IPv4 připojení, ale pro využití tunelu musí mít poté konfiguraci i pro IPv6, zařízení uživatele poté funguje jako dual stack. Na rozdíl od tunel serveru slouží pro komunikaci s uživatelem a realizaci jeho potřeb, respektive tunelu a z technického pohledu se jedná webový server, který musí mít konektivitu

minimálně do IPv4 sítě, po které dochází k výměně dat s uživatelem. Jelikož tunel server je nakonfigurovaný pro oba protokoly, výměna dat mezi brokerem a serverem může probíhat na jednom z nich. Záleží na tom, zda je broker nakonfigurovaný i pro IPv6 a jaký protokol pro komunikaci zvolí provozovatel tunel server/brokeru. Jeden broker má k dispozici několik tunel serverů, což zajistí vyšší efektivitu při správě tunelů. Pro představu se můžeme podívat na Obrázek 9, který nastiňuje princip mechanismu tunel server/broker.



Obrázek 9: Tunel server/broker

Fungování tunel server/brokeru vypadá následovně. První krok, který uživatel uskuteční, je, že provede registraci u brokera přes webový formulář. Uživatel poté dojedná s brokerem způsob zabezpečení a autorizace, které zabrání neautorizovanému využití služby.

Po dokončení autorizačního procesu poskytne uživatel následující údaje:

- Poskytne IPv4 adresu pro svůj konec tunelu.
- Poskytne doménové jméno pro registraci globální IPv6 adresy svého konce tunelu v DNS.
- Poskytne informaci, zda se jedná o samostatného hosta nebo směrovač.

Jde-li o směrovač, tak ještě poskytne informaci, jaký objem IP adres bude potřebovat. Na základě těchto informací a vnitřních pravidel pro řízení provozu vybere nejvhodnější tunel server. Dále zvolí prefix sítě pro uživatele. Nejčastěji se setkáme s prefixem /48 pro celou síť, /64 pro podsíť nebo /128 pro samostatného hosta. Poté se ještě nastaví životnost tunelu. Pak už jen broker odešle vybranému serveru konfiguraci pro vytvoření tunelu na jeho straně a odešle konfiguraci i uživatel. Uživatel provede konfiguraci na své straně, v tu chvíli vznikne tunelované spojení a může komunikovat i po IPv6. (Steffann et al. 2013, Durand et al. 2001, Hagen 2006 s. 271 – 273).

5.2.2 6to4

Tento mechanismus je všeobecně známý pod názvem 6to4, ale jeho primární název je Connection of IPv6 Domains via IPv4 Clouds, jak je uvedeno v jeho specifikaci v RFC 3056. 6to4 je jedním ze zástupců automatického tunelování. Slouží především pro propojení dvou izolovaných IPv6 sítí, jenž mají konektivitu jen do IPv4 internetu, jak ilustruje Obrázek 10. Stačí, když každá izolovaná IPv6 síť, která chce aplikovat 6to4 tunelování, má směrovač, jenž bude zajišťovat provoz 6to4. Je zapotřebí mít minimálně jednu veřejnou IPv4 adresu, tu bude mít 6to4 směrovač, proto také nelze podle RFC 3964 použít následující adresní rozsahy:

- 0.0.0.0/8 (systém nemá IP adresu přiřazenu)
- 10.0.0.0/8 (neveřejné IP adresy)
- 127.0.0.0/8 (loopback)
- 172.16.0.0/12 (neveřejné IP adresy)
- 192.168.0.0/16 (neveřejné IP adresy)
- 169.254.0.0/16 (link-local adresy)
- 224.0.0.0/4 (multicast adresy)
- 240.0.0.0/4 (rezervováno pro budoucí využití)

Zároveň nesmí být použita IP adresa, která je vyhrazena pro broadcast. Tato IPv4 adresa slouží pro výpočet prefixu IPv6 sítě a směrovač jí také vkládá jako zdrojovou adresu do hlavičky IPv4 datagramu, kterým se tuneluje.

Zajímavé je adresování u 6to4. Byl pro to vyhrazen šetnáctibitový prefix 2002::/16. Dalších 32 bitů reprezentuje IPv4 adresu 6to4 směrovače, dohromady to dává prefix /48. Díky tomu zbývá ještě 16 bitů pro vytváření podsítí. Jako příklad, pro výpočet prefixu, má 6to4 směrovač přiřazenou IP adresu 188.16.254.1. Nejprve se převede daná adresa do šestnáctkové soustavy a je připojena k prefixu 2002::/16, tím dojde ke vzniku prefixu 2002:bc10:fe01::/48.

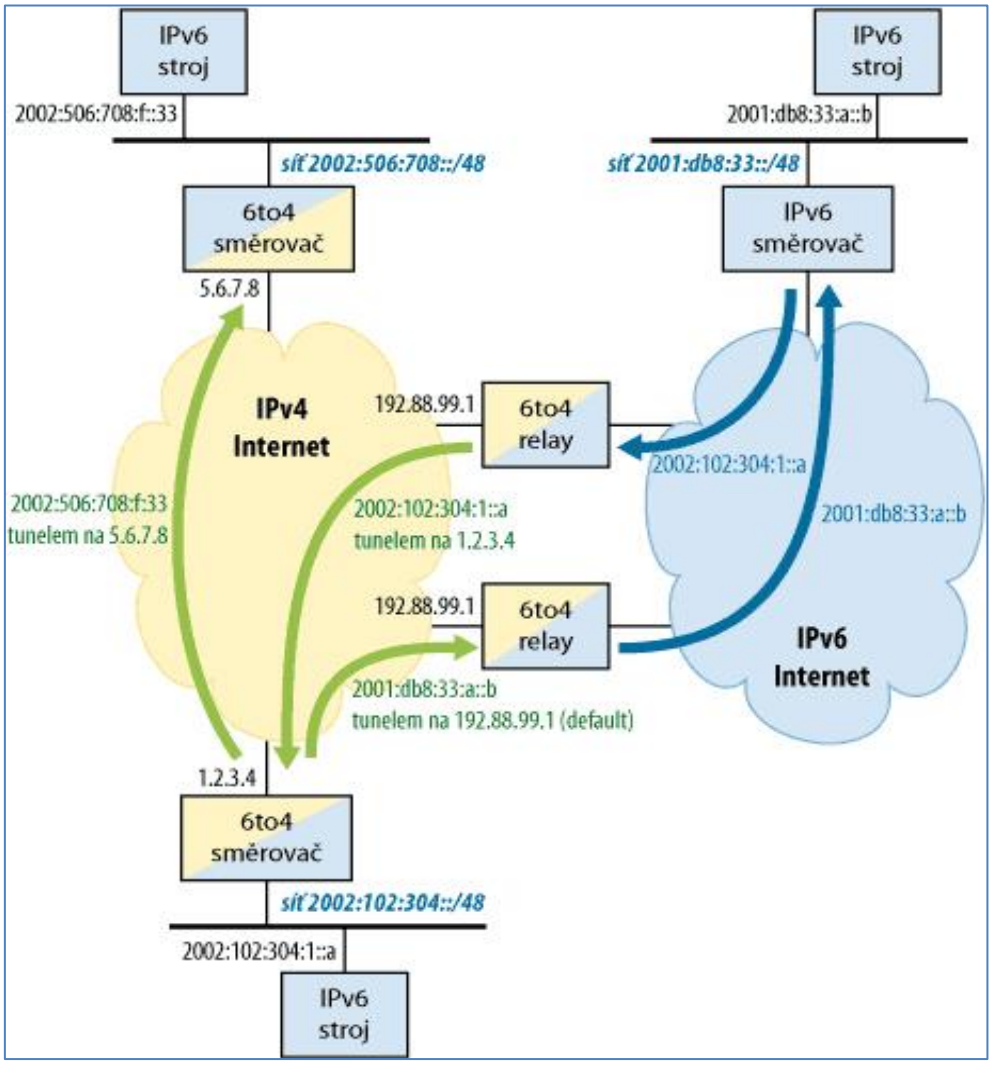
Standardně bývá pro 6to4 vyhrazen hraniční směrovač mezi IPv6 sítí a IPv4 internetem. Na takovém směrovači se provede jen konfigurace pro 6to4, do směrovací tabulky pro IPv6 je přidán záznam a ten stanovuje, že o veškerou komunikaci s prefixem 2002::/16 se odbavuje tento směrovač. Kvůli 6to4 není potřeba provádět IPv4 směrování žádný zásah.

Celé to pak funguje následovně. 6to4 směrovač obdrží IPv6 datagram a cílová adresa obsažená v hlavičce datagramu začíná prefixem 2002::/16, podle toho pozná, že se bude tunelovat. Z cílové IPv6 sítě respektive z následujících 32 bitů po prefixu 6to4 tunelování vypočítá IPv4 adresu 6to4 směrovače cílové sítě. Vytvoří IPv4 datagram, v hlavičce nastaví jako zdrojovou IP adresu svojí IPv4 adresu, jako cílovou IP adresu nastaví vypočtenou IPv4 adresu a do pole protokol nastaví hodnotu 41, že se jedná o tunelovaný datagram. IPv6 datagram pak, stejně jako u předchozích mechanismů, slouží opět jen jako nesená data a směrovač podle tunelovacích datagram pravidel odešle. IPv4 datagram dorazí na 6to4 směrovač, ten z něj vybalí původní IPv6 datagram a směruje ho příjemci. Odpověď probíhá stejným způsobem, kde IP adresa původního odesílatele figuruje naopak jako cílová adresa.

(Carpenter a Moore 2001, Savola et al. 2004, Hagen 2006 s. 264 – 266)

Doposud šlo o variantu, kdy, pomoci 6to4, spolu komunikují dvě izolované IPv6 sítě. 6to4 je možné využít ale i pro komunikaci s nativní IPv6 sítí neboli s IPv6 internetem, viz Obrázek 10. Pro komunikaci mezi 6to4 a nativní IPv6 vznikl zprostředkovatel (relay směrovač). Zprostředkovatel je směrovač, který je nakonfigurovaný pro 6to4 a má přímou konektivitu s nativní IPv6. Je používáno standardní směrování, jen relay směrovač má záznam v IPv6 směrovací tabulce, že skrze něj vede cesta do sítí využívajících 6to4 tzn. komunikace určená pro adresy s prefixem 2002::/16.

6to4 směrovač potřebuje znát IP adresu relay směrovače, aby mohl rozhodnout, kam komunikaci směrovat a to je značný problém. Vzniklo však prosté řešení, kdy všichni zprostředkovatelé mají nastavenu anycast adresu 192.88.99.1. Pak tedy nezbývá nic jiného, než nastavit na 6to4 směrovači implicitní IPv6 cestu k zprostředkovateli, tzn. na adresu 2002:c058:6301::. Dorazí-li na 6to4 směrovač IPv6 datagram s cílovou nativní IPv6 adresou, směrovač vypočítá z adresy 2002:c058:6301:: IPv4 adresu zprostředkovatele. Datagram zabalí, odešle ho na tuto IPv4 adresu a pomocí běžných pravidel je datagram směrován k nejbližšímu zprostředkovateli, který z datagramu vybalí IPv6 datagram a odešle ho do nativní IPv6 sítě. A tady právě vzniká problém s asymetrií směrování, jelikož nejbližší zprostředkovatel při odeslání požadavku nebude stejný jako při odeslání odpovědi. (Huitema a Microsoft Corporation 2001, Hagen 2006 s. 264 – 266)



Obrázek 10: 6to4 [19]

Hlavní výhoda je jednoduchost a nenáročnost na výkon, zejména protože směrování probíhá standardně, jen IPv6 směrovací tabulka projde mírnou úpravou, pouze se provede krátká konfigurace 6to4. Nevýhodou je nespolehlivost 6to4. Jednak je tu problém s firewally, které blokují datagramy, kvůli hodnotě 41 v poli protokol. Firewally rovněž mohou mít problém s asymetrickým směrováním. Každá IPv6 síť využívá pro 6to4 tunelování jiného zprostředkovatele, je tedy téměř jisté, že komunikace jedním směrem nebude směrována stejnou cestou jako komunikace opačným směrem. To vše vede ke ztrátovosti paketů. Jak uvádí Satrapa (2012, s. 261) 10% – 15% požadavků skrze 6to4 tunelování je bez odpovědi např. podle měření RIPE NCC a jiných. Jako náhrada za 6to4, která má odstranit jeho problémy, byl vyvinut mechanismus 6rd. (Satrapa, 2012, s. 261)

5.2.3 6rd

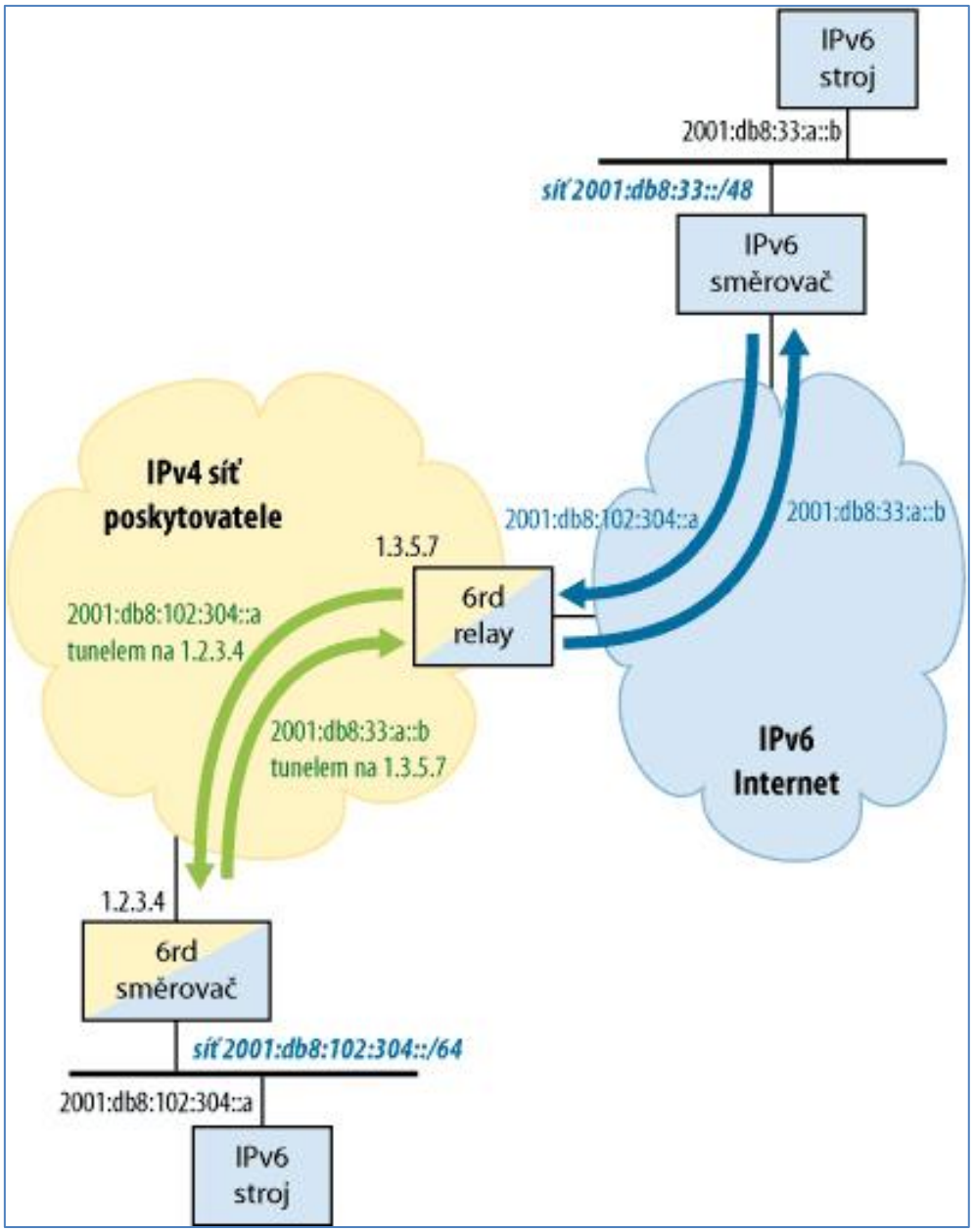
6rd neboli IPv6 rapid deployment je dalším zástupcem automatického tunelování a stejně jako předchozí slouží pro tunelování IPv6 komunikace skrz IPv4 síť. Vychází ze 6to4, ale výrazně se liší v tom, že je nasazeno lokálně u poskytovatele internetu. Veškerá správa přechází do rukou poskytovatele internetu. Jeho páteřní síť funguje stále na IPv4, jen je zapotřebí mít konektivitu do IPv6 internetu. O směrování do a z IPv6 internetu se stará 6rd relay směrovač, jenž musí být, kromě IPv6, nakonfigurován i pro IPv4, po které přijímá tunelované datagramy. Každý uživatel má pak směrovač, na kterém je nastavené 6rd a z tohoto důvodu jsou většinou uživatelské směrovače pod správou poskytovatele internetu. Musí mít také nakonfigurovány oba internetové protokoly, jelikož jsou připojené do páteřní IPv4 sítě ISP a uživatelé mají svojí IPv6 síť.

Stejně jako u 6to4 je adresa tvořena z prefixu mechanismu respektive 6rd a IPv4 adresy směrovače. Maximálně prvních 32 bitů adresy tvoři 6rd prefix poskytovatele internetu a dost často bývá shodný s IPv6 prefixem, který mu byl přidělen. Následujících 32 bitů adresy tvoří IPv4 adresa směrovače. Záleží na možnostech poskytovatele, jakou zvolí délku 6rd prefixu, od toho se pak odvíjí možnost tvorby podsítí. V praxi to téměř vždy funguje tak, že 6rd prefix má 32 bitů. Pro uživatele to znamená, že po přičtení IPv4 adresy jeho směrovače získá IP adresu své sítě s prefixem /64, což mu neumožní vytvářet podsítě.

Poskytovatel internetu má např. prefix 2001:db8::/32 a uživatelův směrovač má IPv4 adresu 192.168.100.10. Po sloučení prefixu s IPv4 adresou směrovače v hexadecimálním tvaru je výsledná adresa sítě 2001:db8:c0a8:640a::/64.

Obrázek 11 demonstruje průběh směrování s nasazeným 6rd. Směrování tunelovaných datagramů ven do IPv6 internetu probíhá standardně, podle směrovacího protokolu, který poskytovatel internetu implementoval na 6rd relay směrovači a příchozí datagramy opačně z IPv6 internetu se posílají tunelem na IPv4 adresu uživatelova směrovače.

Průběh je následující. Uživatel odešle IPv6 diagram, který dorazí na jeho směrovač. Pokud má cílová IPv6 adresa 6rd prefix jeho poskytovatele internetu, z následujících 32 bitů po 6rd prefixu vypočítá IPv4 adresu směrovače. IPv6 datagram zabalí do IPv4 datagramu a odešle. Ten pak putuje pouze IPv4 páteřní sítí poskytovatele k cílovému směrovači, který vybalí IPv6 datagram a směruje ho adresátovi. V jiném případě má jako cíl jinou IPv6 adresu, směrovač rovněž zabalí datagram do IPv4 datagramu a odešle ho IPv4 tunelem na adresu 6rd relay směrovače. Ten vybalí IPv6 datagram a směruje ho do IPv6 internetu na základě cílové IP adresy obsažené v hlavičce. Opačně dorazí datagram v IPv6 internetu na 6rd relay, z cílové adresy získá IPv4 adresu směrovače cílové sítě, vytvoří IPv4 datagram, do něj vloží příchozí IPv6 a odešle ho. 6rd směrovač uživatele tento datagram příjme, vybalí z něj původní a směruje ho adresátovi.



Obrázek 11: 6rd [19]

Hlavní výhodou 6rd je, že tunelování probíhá lokálně v rámci sítě ISP. Komunikace tím předvídatelná, zároveň je symetrická, jelikož cesta tunelem z a do IPv6 internetu je stejná. Další výhodou, která vyplývá z lokálního tunelování, je že na rozdíl od 6to4 odpadá povinnost mít veřejnou IPv4, tudíž ISP může mít IPv4 páteřní síť postavenou na privátních IP adresách. Navíc implementace není náročná a zásah do již fungující IPv4 sítě je minimální. 6rd s sebou nese i jisté nevýhody. Jednou z nich je, že uživatelé jsou závislí na jejich poskytovatelích internetu, protože 6rd si uživatel nemůže nakonfigurovat jako 6to4, kde se datagramy směrují nejbližšímu zprostředkovateli, ale

musí ho IPS poskytovat. Další potíž pramení z prefixů přidělovaným poskytovatelům internetu. Většinou mají přiřazen prefix /32, což při dalších 32 bitech pro IPv4 adresy znamená, že není možné tvořit další podsítě a uživatel dostane jednu svoji síť. (Despres a RD-IPtech 2010, Satrapa 2010)

5.2.4 ISATAP

ISATAP neboli Intra-Site Automatic Tunnel Addressing Protocol vznikl za účelem propojení samotných IPv6 zařízení, nacházejících se v koncové IPv4 síti, aby mohla spolu komunikovat. Je dalším z automatických tunelovacích mechanismů a umožnuje tunelování v IPv4 sítí, ať už se v ní používají globální nebo privátní IP adresy. Z pohledu mechanismu slouží IPv4 síť jako linková vrstva.

Obdobně jako ostatní tunelovací mechanismy i tento vytváří IPv6 adresy z IPv4 adres. Prvních 64 bitů je libovolných, takže může být správcem sítě daná adresa sítě, nebo může obsahovat 6to4 adresu. Kombinace 6to4 a ISATAP je určitě dobrá, jelikož ISATAP bude zajišťovat tunelování uvnitř sítě a 6to4 bude řídit tunelování skrz IPv4 internet. Následujících 64 bitů je identifikátor rozhraní, který je z poloviny tvořen konstantou a ta je buď 0000:5efe nebo 0200:5efe. Posledních 32 bitů IPv6 adresy reprezentuje IPv4 adresa zařízení. Typ IPv4 adresy také rozhoduje o použité konstantě. Jestliže je IP adresa privátní, použije se 0000:5efe, v opačném případě je IPv4 adresa globální a použije se konstanta 0200:5efe.

Průběh směrování je prostý, pokud je cílem IPv6 adresa se stejným prefixem, jako má IP adresa zařízení které chce datagram odeslat, datagram je odeslán přímo cílovému zařízení. Pokud se jedná o jakoukoliv jinou IPv6 adresu, datagram se odešle na ISATAP směrovač. Směrovač je připojen nativně do IPv6 nebo používá například 6to4 pro směrování datagramů ven.

Adresa sítě je např. 2001:db8:a1:a1::/64 a IPv4 adresa zařízení je 192.168.100.10. Hexadecimální zápis IPv4 adresy je c0a8:640a. Vzhledem k tomu, že je adresa privátní, konstanta je 0000:5efe. Z těchto informací je následně vytvořena adresa 2001:db8:a1:a1::5efe:c0a8:640a. Z důvodu, který hned vysvětlím, ISATAP vytvoří ještě link-local IPv6 adresu fe80::5efe:c0a8:640a.

Jelikož IPv4 nepodporuje unicast, nefunguje zde objevování sousedů. Pro lepší předvídatelnost bylo zavedeno, že každé ISATAP zařízení bude mít link-local adresu složenou z link-local prefixu fe80::/10 a identifikátoru zařízení. To řeší problém s objevováním sousedů, jelikož link-local adresu nemusí zařízení zjišťovat, ale zastupuje jí IPv4 adresa sousedního zařízení.

Je tu ještě jeden problém a to je automatická konfigurace směrovačů a adres. Pro ISATAP bylo vytvořeno PLR neboli potential router list (seznam možných směrovačů). Tento seznam obsahuje IPv4 adresy směrovačů, které mají nakonfigurovaný ISATAP. Zařízení pak posílá požadavky na adresy těchto směrovačů a ohlášení jsou zasílána zpět na IPv4 adresu zařízení. Ohlášení obsahuje ISATAP prefix, ze kterého si zařízení sestaví IP adresu a nakonfiguruje směrování. Otázka je, kde zařízení PLR získá, pokud ho nemá k dispozici. Tento proces nebyl do mechanismu implementován, a proto se vedle manuální konfigurace využívá DNS. Záznam v DNS je vždy označen *isatap,* viz Tabulka 1.

Tabulka 1: ISATAP DNS záznamy

isatap	A	192.168.100.1
	A	192.168.100.64
	A	192.168.100.128

Tento příklad uvažuje tři ISATAP směrovače dostupné na adresách 192.168.100.1, 192.168.100.64 a 192.168.100.128. Zařízení, které chce provést konfiguraci ISATAP se dotáže DNS na příslušné záznamy a po obdržení se dotazuje příslušných směrovačů atd., jak jsme si už popsali. (Templin et al. 2008, Podermański a Grégr 2011)

5.2.5 Další tunelovací mechanismy

Mezi další tunelovací mechanismy patří 6over4, který funguje na totožném principu jako ISATAP. Také slouží pro komunikaci separovaných IPv6 zařízení v koncové IPv4 síti. Na rozdíl od ISATAPu si zakládá na objevování sousedů a dalších standardních mechanismech. Využívá techniky mapování IPv6 multicast adres na IPv4 multicast adresy. Adresy se mapují tak, že prvních 16 bitů je 239.192 a zbytek tvoří koncových 16 bitů IPv6 multicast adresy. (Carpenter et al. 1999)

Důležitým tunelovacím mechanismem je Dual-stack lite (DS lite). Na rozdíl od ostatních, které jsou v této kapitole, má opačnou funkci. Definuje tunelování IPv4 skrz IPv6 síť. Předbíhá dobu, předpokládá vyčerpání IPv4 adres a hojné rozšíření IPv6. Umožňuje pokračování podpory IPv4 a pobízí k nasazení IPv6. Odděluje také implementaci protokolu v síti ISP od internetu, což umožňuje snadnější přechod.

Každý uživatel má svojí IPv4 privátní sít a směrovač s přístupem do nativní IPv6 sítě. Všechna komunikace, kromě té v rámci své privátní sítě, putuje na Address Family Transition Router (AFTR). AFTR je centrální prvek provozovaný ISP, který má konektivitu do IPv4 internetu a směruje komunikaci do i z nativní IPv4 sítě. Zároveň má nakonfigurován NAT, pro překlad privátních IPv4 adres uživatele. AFTR má pro to zvlášť tabulku, ve které je každý záznam rozšířen o IPv6 adresu uživatelova IPv6 směrovače. (Durand et al. 2011)

5.3 Translátory

Doposud byly zmíněny případy, kde buď v podání dvojího zásobníku je na výběr, která verze internetového protokolu je použita pro odeslání datagramu nebo je potřeba doručit komunikaci jedné verze, v našem případě zejména IPv6, skrz síť druhé verze internetového protokolu pomocí tunelování. Nyní však nastává situace, kdy spolu mohou komunikovat zařízení z IPv4 a IPv6 sítě.

Za tímto účelem vznikly takzvané translátory. Mají za úkol překládat adresy a datagramy jedné verze IP na druhou. V jádru jsou stejné a každý má pak svá specifika. V následujících podkapitolách jsou jednotlivě popsány. (Hagen, 2006, s. 278)

5.3.1 SIIT

SIIT nebo taky Stateless IP/ICMP Translation je původní mechanismus pro bezstavové překládání hlaviček IP datagramů mezi IP verze 4 a 6. Je to předchůdce všech translátorů a ty z něj vycházejí. Překlad hlaviček datagramů je přesný, pravidla jsou omezená, takže rozšíření hlaviček nejsou podporována.

Překlad IP adres se zdá jednoduchý. Je daný prefix /96, který se nemění a zbylých 32 bitů je pro IPv4 adresu. Může být buď daný poskytovatelem internetu, popřípadě i univerzální 64:ff9b::/96. Překlad z IPv4 na IPv6 adresu je poměrně jednoduchý, jelikož IPv6 datagram je větší než IPv4. K /96 prefixu se přidá /32 IPv4

adresa. V opačném případě se spíše využívá dynamické překládání, které je podobné jako u NATu pro překlad privátních IPv4 adres na globální.

Překlad z IPv4 na IPv6 je jednodušší. Data, která jsou transportována v datagramech, neprochází žádnou změnou, pouze se udělá kontrolní součet transportní hlavičky. Změna proběhne u hlaviček, kdy IPv4 hlavičku nahradí IPv6. Naplnění hlaviček je sice prosté, ale může nastat problém, jelikož je fragmentace IPv4 a IPv6 odlišná. Naplnění hlaviček znázorňuje Tabulka 2. Je-li fragmentace vypnutá, pak je IPv4 datagram jednoduše přeložen. Pakliže je zapnutá, omezí se maximální velikost IPv6 datagramů na minimální velikost propustnosti každé IPv6 linky, což je 1280 bajtů.

Tabulka 2: Překlad IPv4 na IPv6

Verze:	6
Třída provozu:	TOS
Značka toku:	0
Délka obsahu:	Celková délka z IPv4 hlavičky - délka IPv4 hlavičky
Další hlavička:	Protokol, hodnotu 1 (ICMPv4) změnit na 58 (ICMPv6)
Max. skoků:	TTL - 1
Adresa odesilatele:	podle mapování
Cílová adresa:	podle mapování

Zdroj: IPv6.cz 2011

Ohledně maximální velikosti IPv6 datagramů je doporučeno konfigurovat parametry pro její výpočet, což by mělo nahradit zbytečné přidávání hlavičky fragmentace, i když datagram není větší než 1280 bajtů.

Překlad z IPv6 do IPv4 postupuje obdobným způsobem. Tabulka 3 ukazuje pravidla pro naplnění hlaviček. Celková délka hlavičky se poníží o 8, jelikož v ní je zahrnuta i osmibajtová fragmentační hlavička. Jelikož je IPv4 fragmentace více benevolentní, není potřeba k ní přihlížet. V případě kdy je fragmentace zakázána a maximální velikost IPv6 datagramu je přesažena, oznámí se pomocí ICMP chyba. (Li et al. 2011, Hagen 2006 s. 278 – 280)

Tabulka 3: Překlad IPv6 na IPv4

Verze:	4
Délka hlavičky:	5
TOS:	Třída provozu
Celková délka:	Délka obsahu z IPv6 hlavičky + 20
Identifikace:	0 (nebo Identifikace z hlavičky Fragmentace)
Příznaky:	MF=0, DF=1 (nebo MF=M z Fragmentace a DF=0)
Posun fragmentu:	0 (nebo Posun fragmentu z Fragmentace)
TTL:	Max. skoků - 1
Protokol:	transportní protokol z IPv6 hlaviček, hodnotu 58 (ICMPv6) změnit na 1 (ICMPv4)
Kontrolní součet:	vypočíst
Adresa odesilatele:	podle mapování
Cílová adresa:	podle mapování

Zdroj: IPv6.cz 2011

5.3.2 Další translátory

NAT64 je zástupcem translátorů. Je zaměřený především na zpřístupnění nativní IPv4 sítě pro koncové IPv6 sítě popřípadě uživatele. Bezstavově mapuje IPv4 adresy na IPv6. Jeho chování je srovnatelné NATy pro IPv4, má také minimálně jednu veřejnou IPv4 adresu a zjednodušeně řečeno, namísto neveřejných IPv4 adres překládá IPv6 adresy. Překládá pouze protokoly TCP, UDP a ICMP. (Bagnulo et al. 2011)

Dalším translátorem je BIH neboli Bump in the Host. Je náhradou za původní mechanismy BIS a BIA. Soustřeďuje se na koncové počítače, jež jsou připojené jen do IPv6 sítě, kde má zajistit překlad z IPv4 do IPv6 pro aplikace, které IPv6 nepodporují. Jeho výhoda je, na rozdíl od NAT64, možné použití privátní IPv4 adresy. BIH je náročný na konfiguraci a bude třeba člověka zkušeného v IT, což je nevýhodou, když je určen pro koncové počítače. (Huang et al. 2012)

Toto samozřejmě nejsou jediné translátory, další alespoň jmenovitě NAT-PT, TRT a jiné.

6 Praktická část

Tato část práce je věnovaná mechanismům v praxi. Záměrem je prozkoumat, jak se může běžný uživatel připojit do IPv6 světa, když jeho ISP nabízí pouze IPv4 konektivitu. V této situaci se nabízí tunelování nebo využití nějakého translátoru. Jelikož je tato práce, mimo ostatním mechanismům, nejvíce věnovaná tunelování, praktická část je zaměřena právě na to. Také konfigurace automatického tunelu je snadnější a funguje zde i lepší podpora, ostatně např. Teredo je implementováno přímo v operačním systému Windows.

Spousta uživatelů má kabelové nebo bezdrátové připojení od místního poskytovatele internetu a ti se do IPv6 příliš nehrnou. Navíc většina uživatelů má obyčejný směrovač pro malou domácí síť bez podpory nové verze internetového protokolu. I když není k dispozici vhodný směrovač, je možné využít tunelování pro připojení do IPv6 světa.

Známým poskytovatelem tunelů je Hurricane Electric (*www.he.net*), který dokonce spolupracuje, i v Čechách, s datacentrem CE Colo Czech s.r.o. se sídlem v Praze. Avšak je zde jeden problém s tunelem od Hurricane Electric. Pro využívání jejich tunelu je veřejná IPv4 adresa podmínkou. Toto řešení nebude předmětem zkoumání této práce, jelikož je zaměřena na nejčastější situaci, kdy koncové zařízení je za dvojitým NATem. První překlad adres je na směrovači uživatele a druhý na straně ISP.

Jako další uvažované řešení se nabízelo použití tunelu služby SixXS (*www.sixxs.net*) s využitím protokolu Anything In Anything pro překonání NATu. Bohužel 6. 6. 2017 byly veškeré služby ukončeny a projekt SixXS zanikl. Shodou okolností SixXS mělo v České republice v provozu server u firmy IGNUM s.r.o., která rovněž sídlí v Praze.

Následující varianta je tunelovací mechanismus Teredo. Další podkapitoly se zabývají jeho teoretickým popisem a následovně konfigurací a testováním.

6.1 Teredo

Teredo je významné tím, že umožňuje tunelování i přes několikanásobný překlad adres (NAT) a odpadá potřeba mít veřejnou IPv4 adresu. To platí v případě, kdy se jedná o trychtýřový (cone) nebo omezený NAT, se symetrickým si bohužel Teredo neporadí. Jak už bylo zmíněno, tento tunelovací mechanismus je navíc implementovaný

přímo v operačních systémech Windows. Samozřejmě je dostupné i pro Linux, MacOS a další v podobě softwaru Miredo, který je potřeba doinstalovat. Vyhlídky jsou potěšující, ale bohužel Teredo trpí velkou nespolehlivostí a není příliš efektivní.

Podobně jako u 6to4 je i pro tento mechanismu vyhrazen prefix a to 2001::/32. Následujících 32 bitů obsahuje IPv4 adresu Teredo serveru. Zbývajících 64 bitů je určeno na straně uživatele. Prvních 16 bitů z druhé poloviny adresy obsahuje příznaky. Za zmínku stojí příznak C, který nabýval hodnoty 0 nebo 1, podle toho, zda uživatel byl za trychtýřovým NATem (hodnota 1) nebo jiným typem (hodnota 0). Od tohoto řešení se upustilo, jelikož jakékoliv zařízení mohlo zjistit z IP adresy, jaký typ překladu adres se používá a to představovalo bezpečnostní riziko. Proto bylo vydáno RFC 5991, ve kterém je dáno, že hodnota příznaku bude vždy 0. Dalších 16 bitů obsahuje uživatelův UDP port. Zbylých 32 bitů obsahuje IPv4 adresu směrovače s nakonfigurovaným překladem adres, za kterým se nachází zařízení uživatele.

Práce Tereda vypadá následovně. Pozornost je soustředěna rovnou na proceduru, kterou nastavilo RFC 5991, kdy se vždy automaticky počítá s tím, že se jedná omezený NAT. Uživatel na začátku nemá IPv6 adresu a tu získá od Teredo serveru. Teredo server musí být takové zařízení, které má nakonfigurováno IPv4 i IPv6. V operačních systémech Windows je jako výchozí *teredo.ipv6.microsoft.com*, což je server provozovaný společností Microsoft. Pro uživatele v ČR je lepší z hlediska efektivnosti využít server *teredo.nic.cz* provozovaný sdružením CZ.NIC. Uživatel zašle požadavek serveru a datagram obsahuje jeho IPv4 adresu NATu a UDP port. Odpověď s IPv6 prefixem a ostatními informacemi se vrátí ze stejné IPv4 adresy, na kterou byla žádost zaslána a ta projde NATem zpět k němu. Znovu zašle požadavek, ale na druhou adresu serveru. Server na základě informací z obou požadavků vyhodnotí, o jaký překlad adres se jedná. Pokud se informace shodují, jde o NAT omezený nebo trychtýřový. Teredo server pošle odpověď uživateli a ten si podle obdržených informací sestaví IPv6 adresu. Pokud se informace neshodují, pak je to NAT symetrický a není možné toto tunelování použít.

Samotná komunikace probíhá takto. Pokud příjemce také používá Teredo, odešle uživatel dva prázdné datagramy tzv. bubliny. Jednu pošle přímo příjemci, který jí sice zahodí, ale v NATu odesilatele se vytvoří cesta zpět. Druhou bublinu pošle přes Teredo server, který bublinu přepošle příjemci. Druhá bublina příjemci dorazí, jelikož v jeho

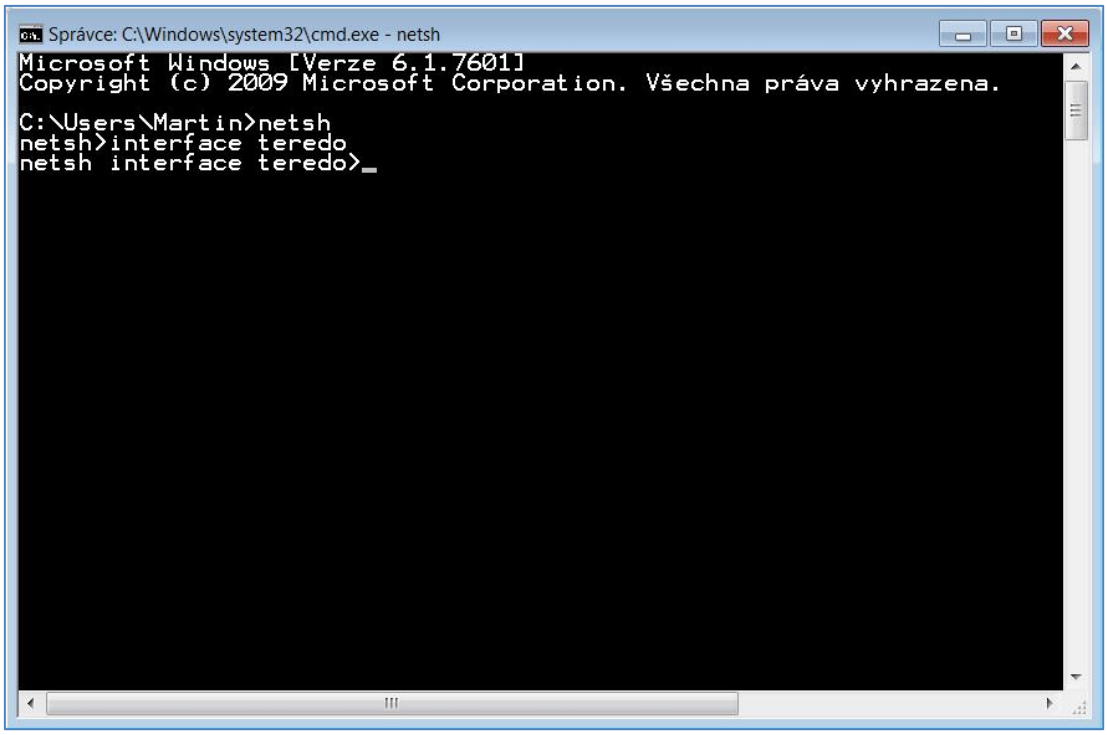
NATu již existuje cesta pro komunikaci z tohoto směru. Příjemce pak odešle bublinu přímo odesilateli a tím se vytvoří cesta pro komunikaci i v jeho NATu.

Pro komunikaci mezi Teredo a IPv6 internetem je zapotřebí ještě zařízení zvané Teredo relay. Je to směrovač připojený do světa obou verzí IP. Zároveň směrem do IPv6 vysílá informaci o tom, že skrze něj je přístupné Teredo. Pokud tedy uživatel chce komunikovat se zařízením v nativním světě, musí nejdříve poslat ICMPv6 zabalené v IPv4 datagramu na Teredo server. Ten pak vybalí ICMPv6 a pošle příjemci. Příjemce pošle odpověď na nejbližší relay směrovač. Ten odešle odpověď přes server odesílatele. Odesilatel z odpovědi zjistí potřebné informace pro komunikaci s daným relay směrovačem. Následně odešle bublinu relay směrovači, tím se vytvoří cesta skrz NAT a je možné komunikovat přímo. (Huitema a Microsoft Corporation 2006, Thaler et al. 2010)

6.2 Konfigurace

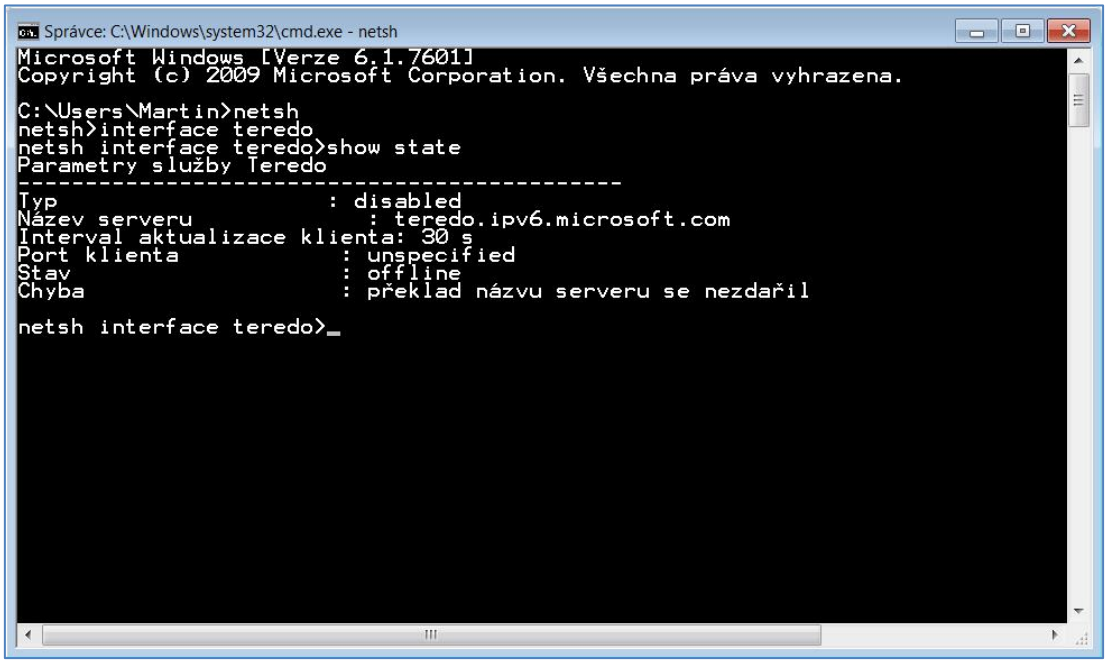
Tolik k teoretickému popisu Tereda, teď přichází na řadu praxe. Konfigurace je prováděna v operačním systému Windows 7. V nastavení síťové karty jsou povolené obě verze internetového protokolu. Mechanismus je již nainstalovaný a připravený ke konfiguraci. Použity jsou dva, již zmíněné, Teredo servery, první od společnosti Microsoft *teredo.ipv6.microsoft.com* a druhý provozovaný sdružením CZ.NIC *teredo.nic.cz*.

Konfigurace probíhá z valné většiny v příkazovém řádku. První je použit příkaz *netsh*, který slouží pro práci se síťovým připojením. Příkazem *interface teredo* se zvolí právě rozhraní Teredo, viz Obrázek 12.



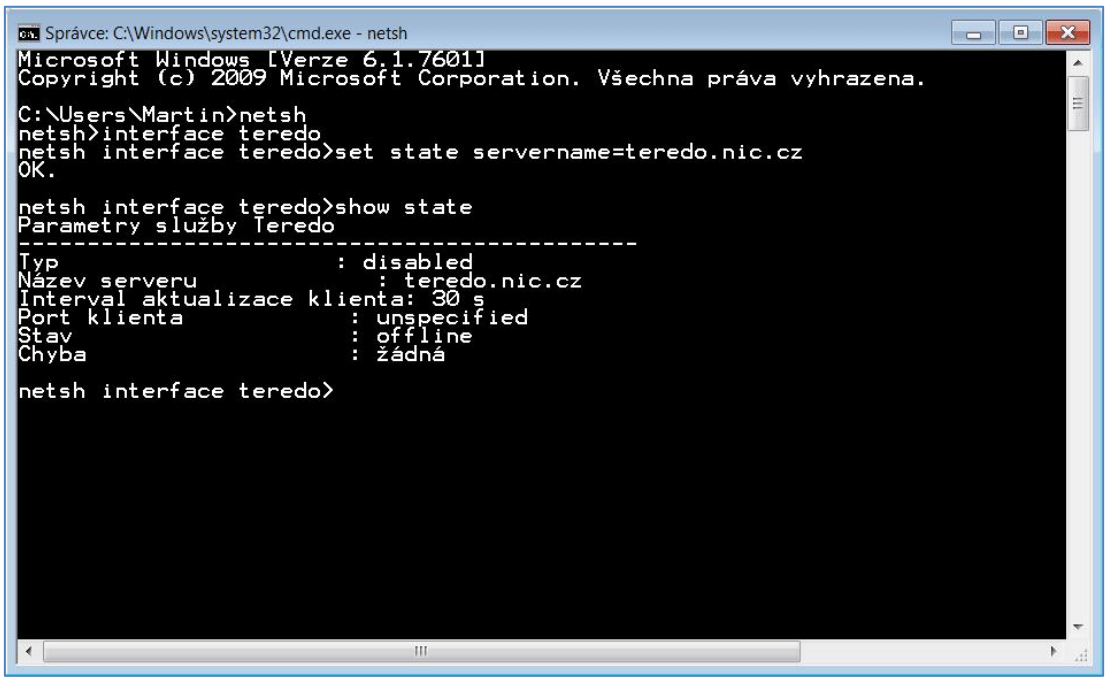
Obrázek 12: Přístup ke konfiguraci Teredo rozhraní

Nejprve je potřeba ověřit stav příkazem *show state*. Na Obrázku 13 můžeme vidět, že je Teredo je zakázané a jako výchozí Teredo server je nastavený již zmíněný *teredo.ipv6.microsoft.com.*



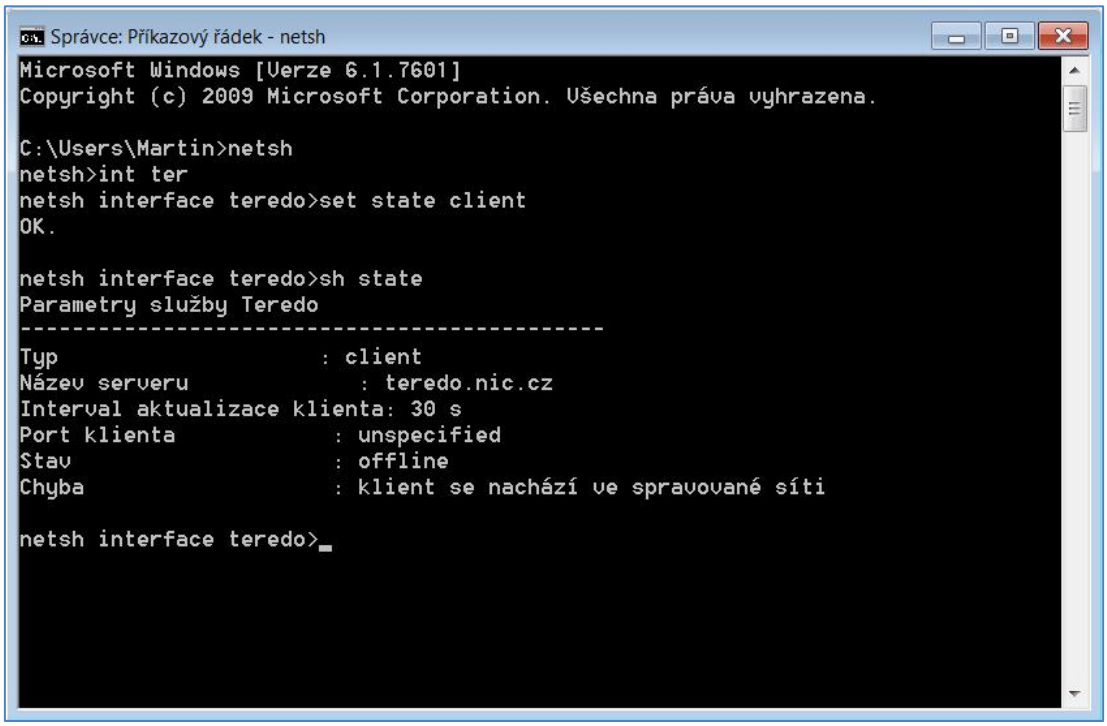
Obrázek 13: Zobrazení stavu služby Teredo - výchozí stav

Z jistého důvodu, který je později zmíněn v testování, je potřeba změnit server. Slouží k tomu příkaz *set state servername=“nazevServeru“*. Jak je vidět na Obrázku 14, je aplikován server od CZ.NIC *teredo.nic.cz*. Následně je dobré ověřit, zda byla změna provedena stejně jako v předchozím případě.



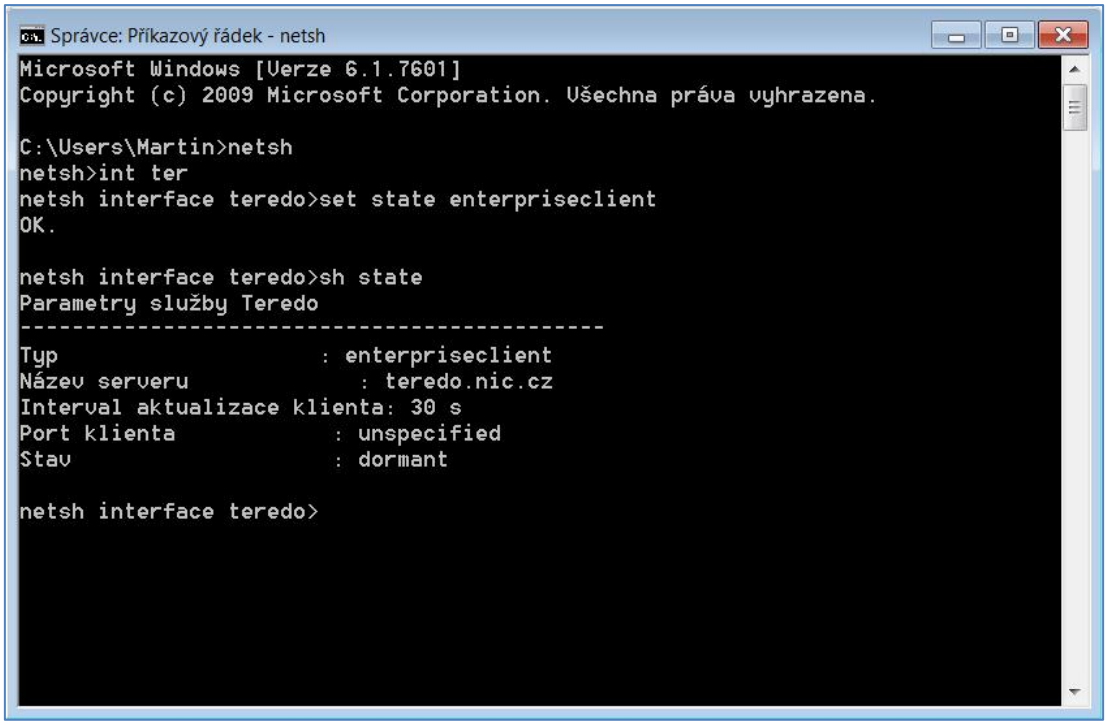
Obrázek 14: Změna Teredo serveru

Teredo je nastavené, ale služba není stále aktivní. Příkazem *set state client* se provede aktivace a typ klienta se nastaví na *client*, viz Obrázek 15.



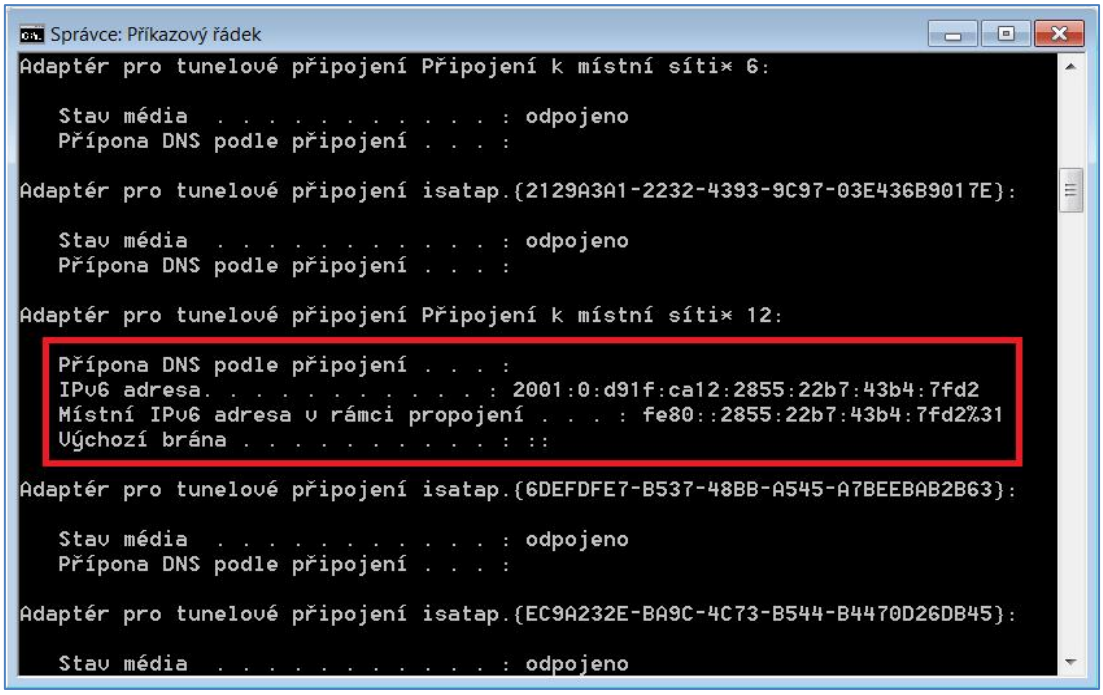
Obrázek 15: Změna typu klienta 1

Ihned je z Obrázku 15 zřetelné, že nastal problém a služba fungovat nebude. Typ klienta *client* se dá použít jen v případě, že zařízení není ve spravované síti. V případě že se v ní nachází, je Teredo klient automaticky vypnut. Řešení je, viz Obrázek 16, změnit stav na *enterpriseclient*, který tuto skutečnost ignoruje.



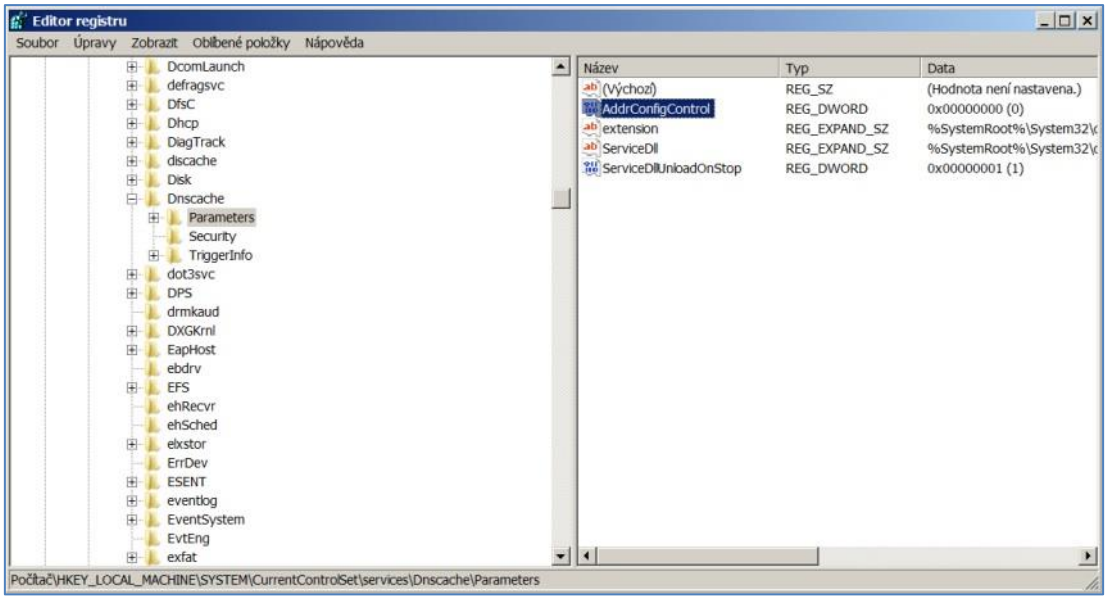
Obrázek 16: Změna typu klienta 2

Stav se změnil z *offline* na *dormant*, což znamená, že ve spolupráci s Teredo serverem byl vytvořen tunel, ale ten ještě nebyl použit. Je potřeba ještě ověřit, zda byla v pořádku vytvořena IPv6 adresa. Stačí pro to použít příkaz *ipconfig*.



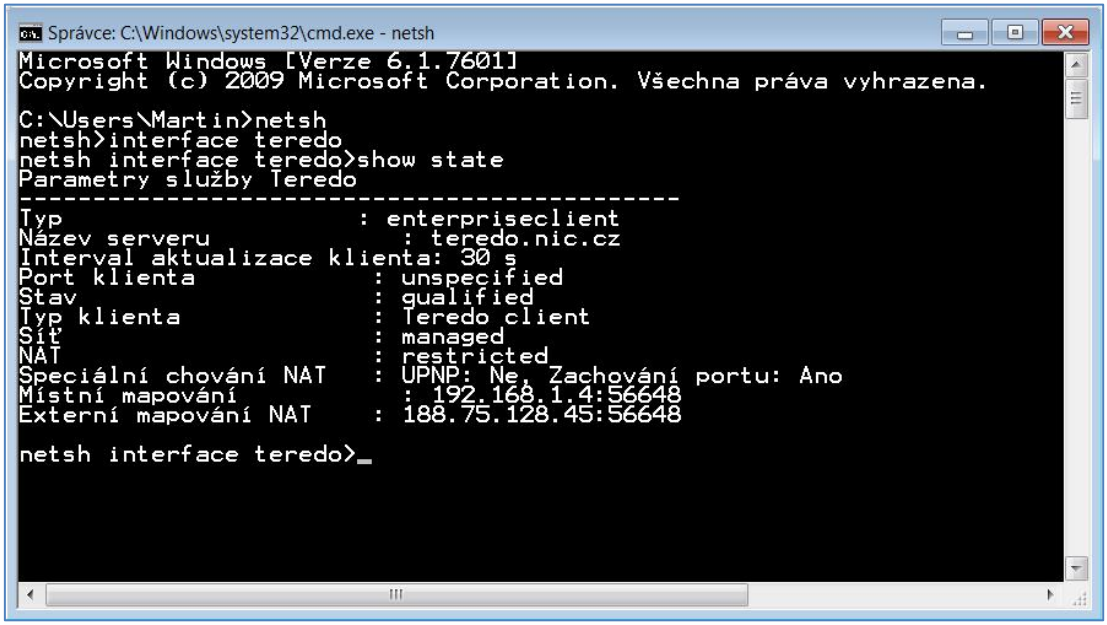
Obrázek 17: IPv6 Teredo adresa

Z Obrázku 17 je evidentní, že byla v pořádku, podle informací od serveru, vytvořena adresa 2001:0:d91f:ca12:2855:22b7:43b4:7fd2. Teredo je spuštěné, adresa nastavena, ale ještě je potřeba zajistit, aby DNS klient neodesílal dotaz DNS serveru jen na A záznam. Děje se to v případě, kdy zařízení má lokální linkovou nebo Teredo IPv6 adresu. „*Toto chování se dá změnit, když do registru HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\Dnscache\Parametrs vložíte hodnotu typu DWORD nazvanou AddrConfigControl obsahující nulu.*“ (Satrapa 2012 s. 58). Pro přidání hodnoty se spustí editor registru, poté jen stačí dohleda registr *Parametrs* a vložit hodnotu. Jak následně vypadá registr, demonstruje Obrázek 18. Výsledek je, že DNS klient bude moci skrze tunel posílat požadavky na AAAA záznamy a díky tomu je možné procházet i weby, které fungují jen na IPv6.



Obrázek 18: Registry

V tuto chvíli je vše nastavené a IPv6 je připraveno k použití. Jakmile je tunel použit, změní svůj stav na *qualified* a zobrazí se i další informace viz Obrázek 19. V následující kapitole se testuje, jak kvalitní bude připojení skrz Teredo.

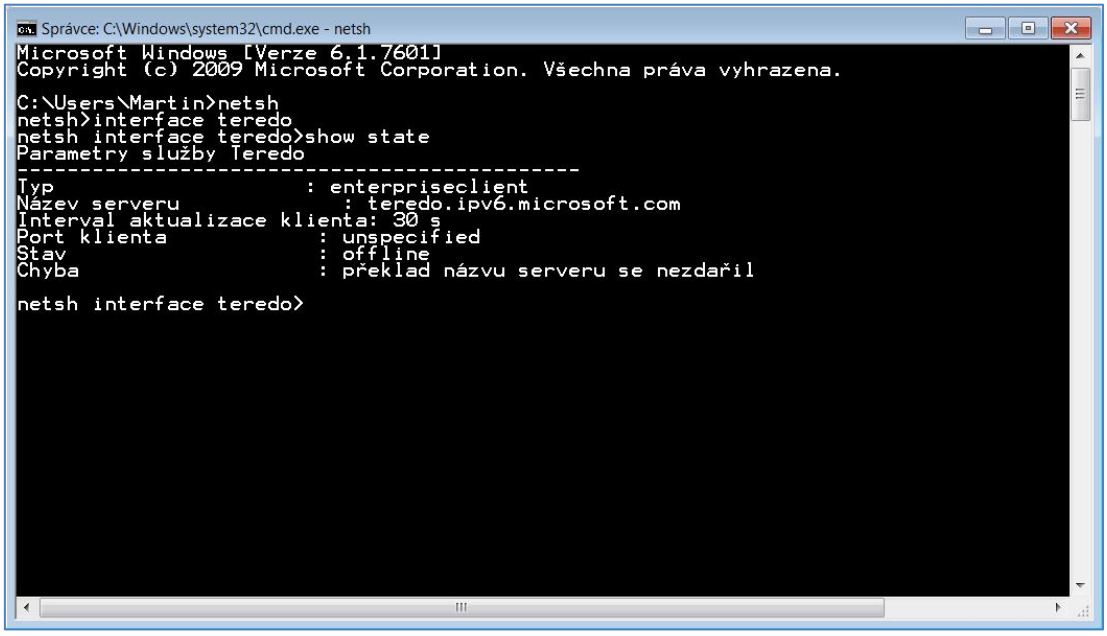


Obrázek 19: Stav Tereda po použití tunelu

6.3 Testování

K testování byly použity základní nástroje pro testování připojení s využitím příkazového řádku. Dále byly provedeny testy skrze webové rozhraní na serverech *www.test-ipv6.cz* a *www.ipv6-test.com*.

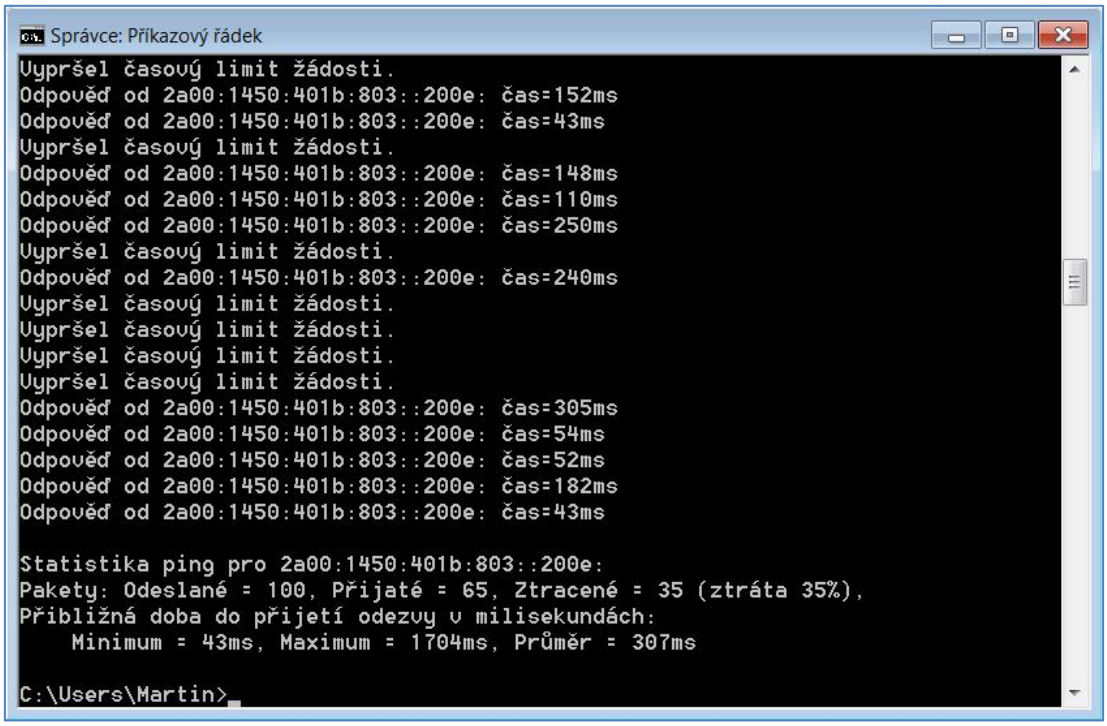
Bohužel ještě před samotným testováním se projevilo, že výchozí Teredo server *teredo.ipv6.microsoft.com* není možné použít. Jak naznačuje Obrázek 23, není možné ho použít. Je zobrazena chyba, nezdařil se překlad názvu severu. Příčinou je jeho nedostupnost. Nejedná o chybu mechanismu, jak dokazují následné testy, ale problém je na samotném serveru. V řadě diskuzí zaměřených na problematiku Tereda na různých fórech se uživatelé zmiňují, že má časté výpadky či je zcela nedostupný. Z tohoto důvodu pro veškeré testování proběhne s využitím českého Teredo serveru *teredo.nic.cz*.



Obrázek 20: Nepoužitelný Teredo server

6.3.1 Test č. 1

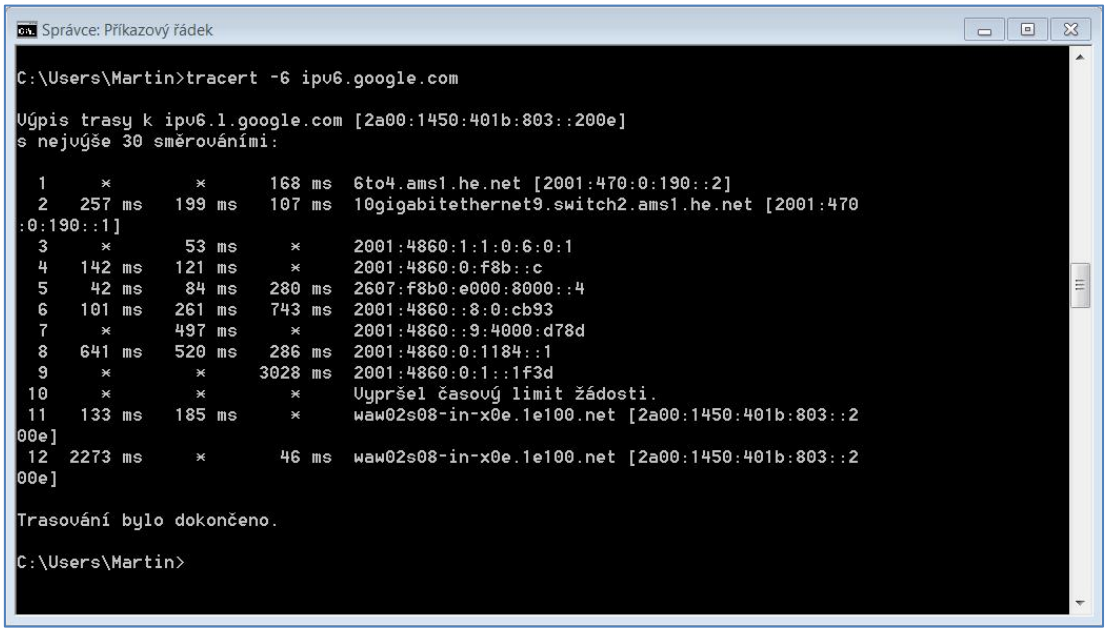
Pro první test poslouží opět příkazový řádek a nejzákladnější diagnostický nástroj. Otestuje se odezva vůči serveru *ipv6.google.com*. Viz obrázek 21, diagnostika je spuštěna příkazem *ping -6 ipv6.google.com -n 100*. Parametr *-6* určuje použití protokolu IPv6 a *-n 100* říká, že se odešle sto ICMPv6 paketů.



Obrázek 21: Ping

6.3.2 Test č. 2

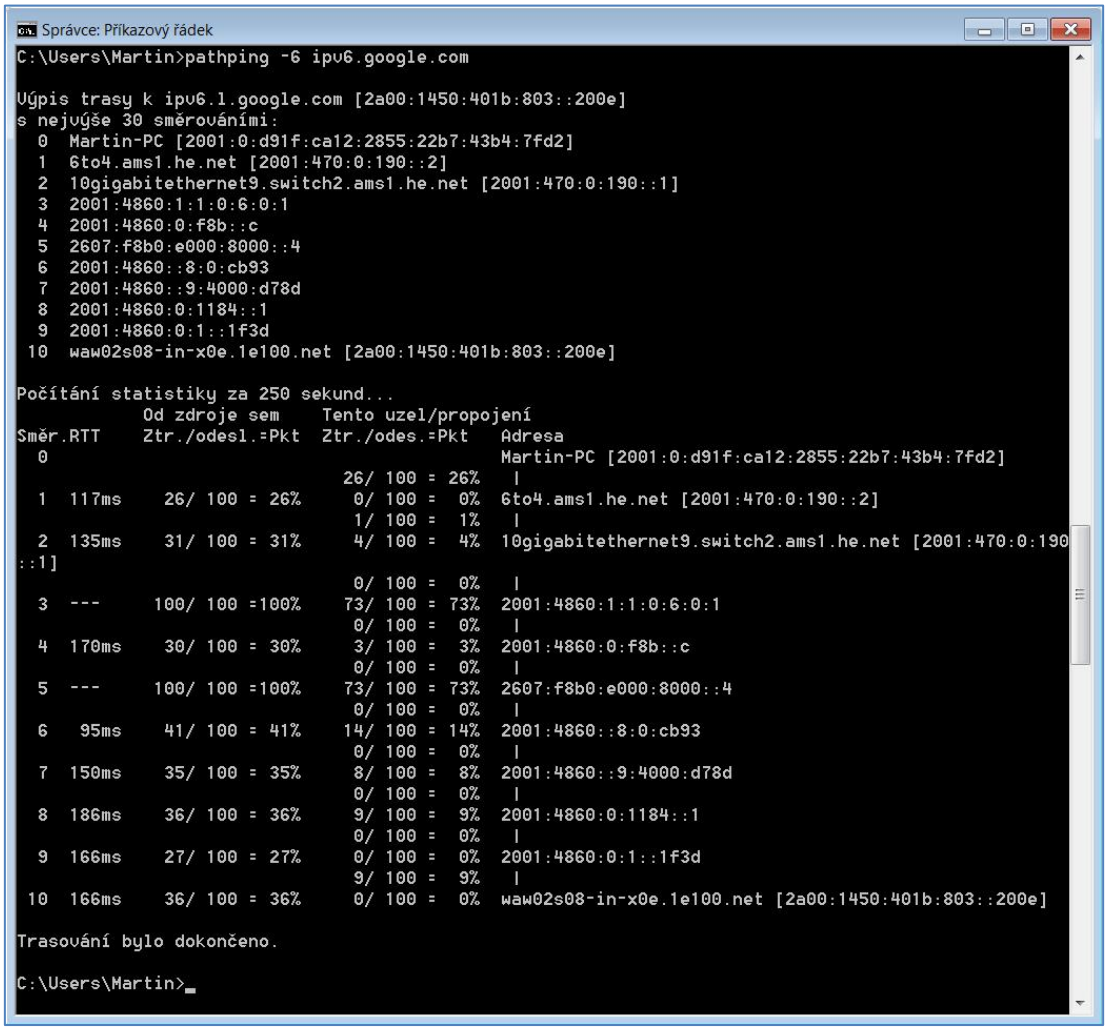
Obdobným způsobem jen proveden následující test. Teredo je podrobeno diagnostice trasování na stejný server pomocí *tracert -6 ipv6.google.com*. Zde je použit pouze parametr *-6*, protože se jedná o IPv6. Výsledek je zobrazen na Obrázku 22.



Obrázek 22: Trasování

6.3.3 Test č. 3

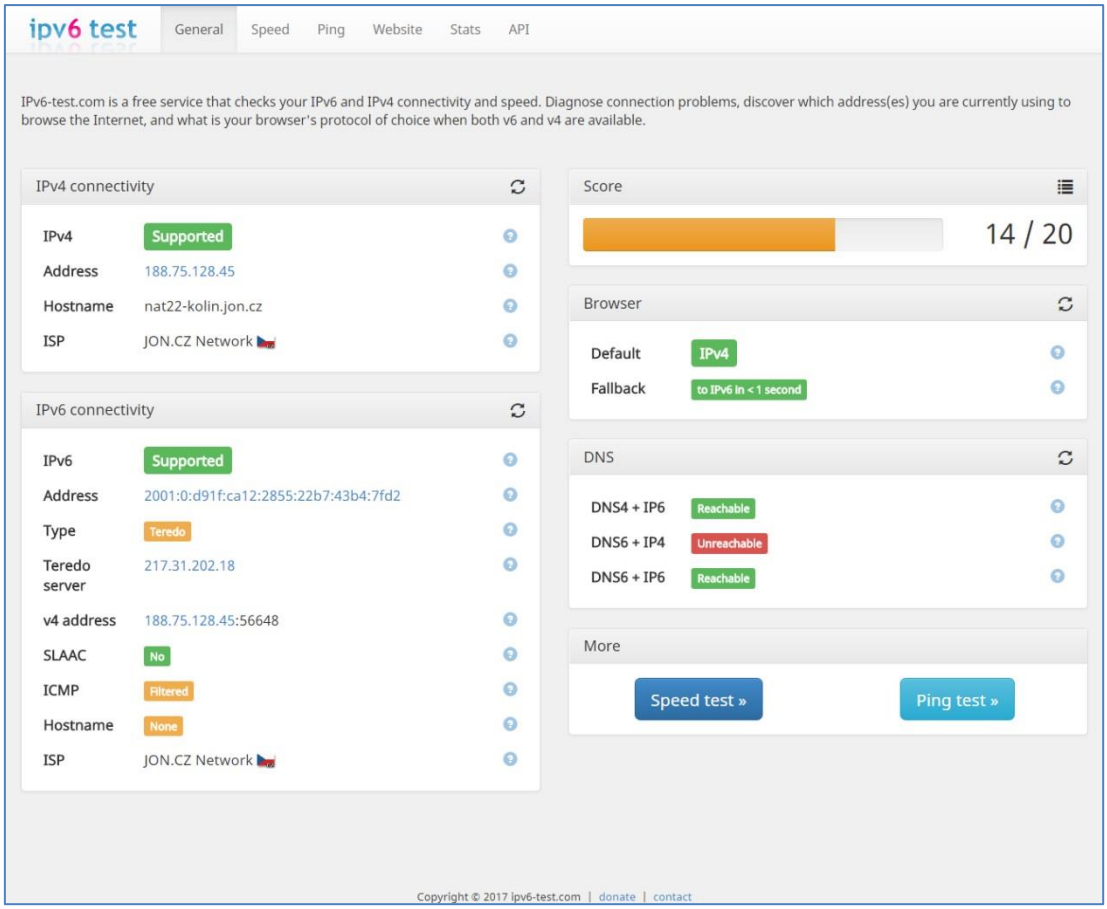
Poslední zkouškou v příkazovém řádku bude tzv. path ping. Je to rozšířený nástroj, který kombinuje trasování a testu odezvy, jak je vidět na Obrázku 23. Příkaz je téměř shodný s trasováním. Pro IPv6 je to *pathping -6 ipv6.google.com*.



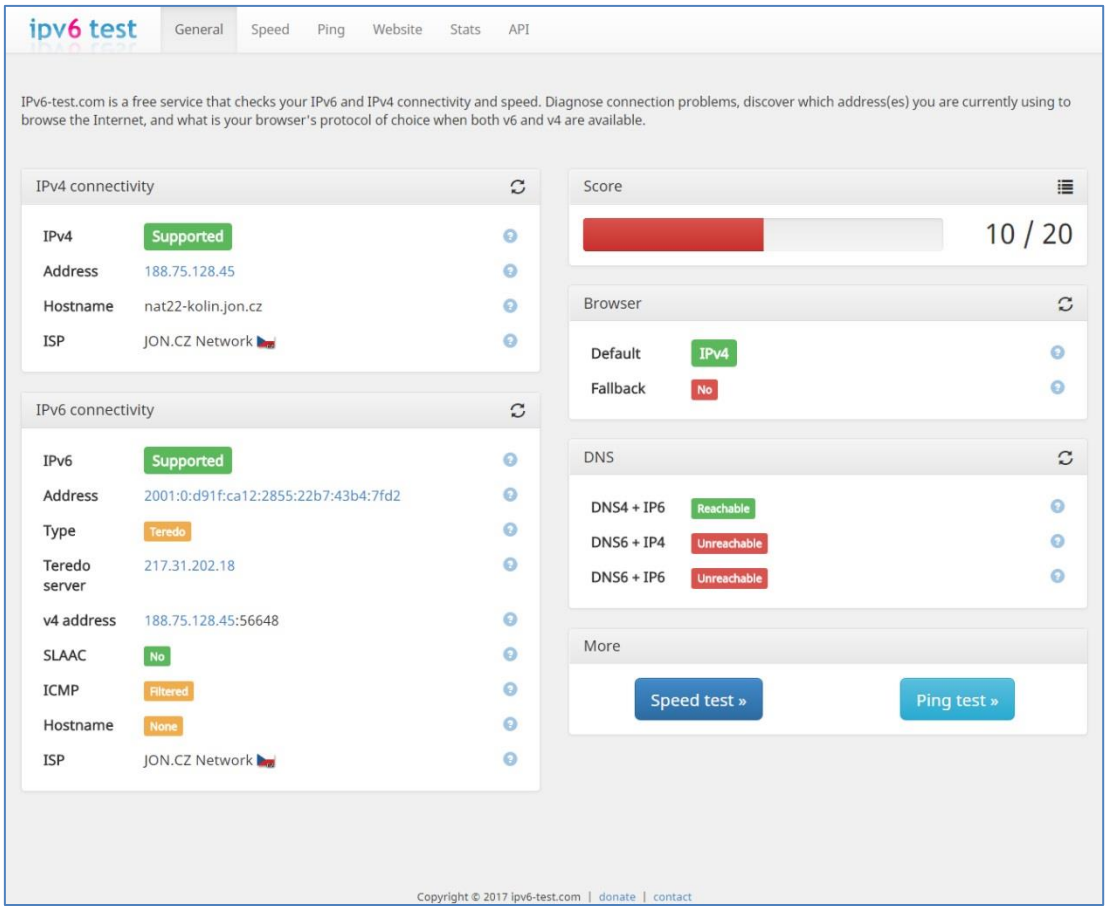
Obrázek 23: Path ping

6.3.4 Test č. 4

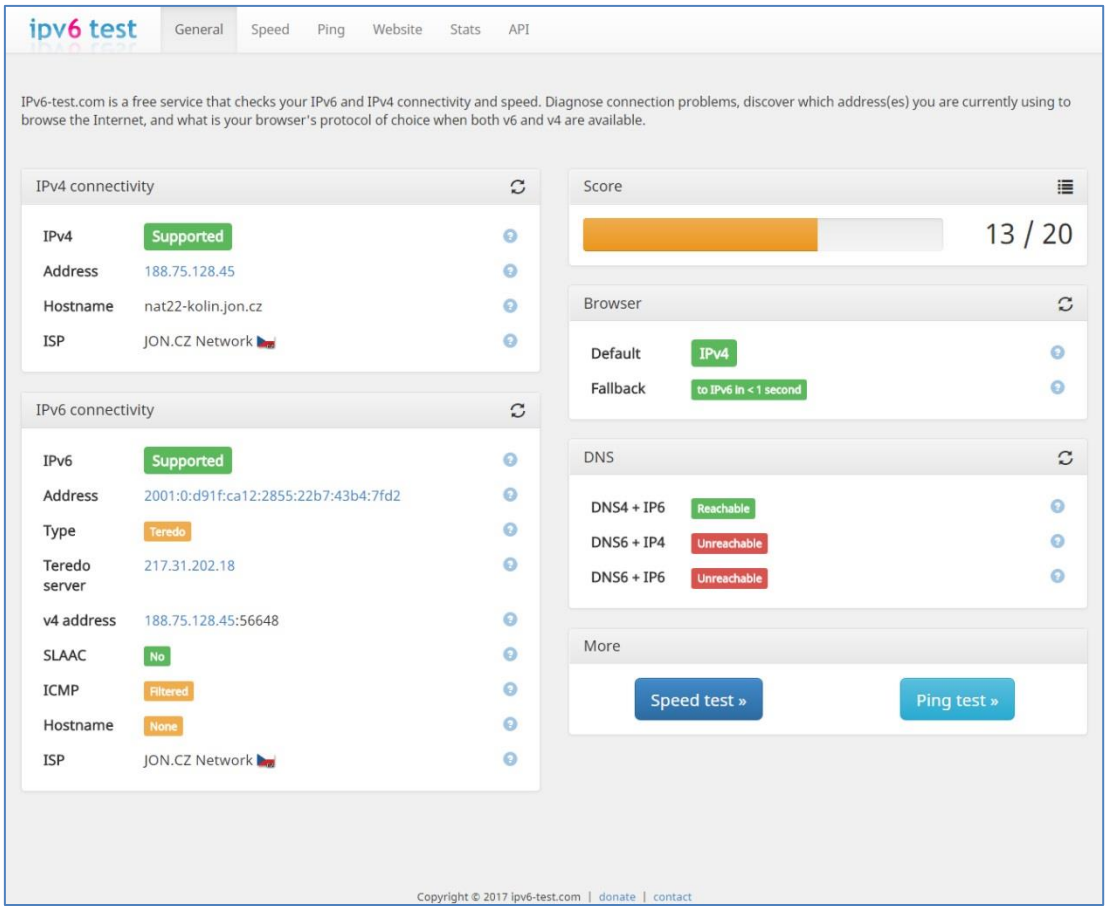
Pro čtvrté testování je využita služba IPv6-test.com. Je to bezplatná online služba pro testování IPv4 a IPv6 připojení dostupná na webu *www.ipv6-test.com*. Hned po načtení stránky se spustí test automaticky. Služba zkontroluje konektivitu pro oba protokoly. Zjistí používané IP adresy a to včetně Tededo serveru. Ověří se DNS. Nejdříve dostupnost IPv6 adresy skrze DNSv4, poté dostupnost adres obou verzí IP skrze DNSv6. Služba umí zjistit také, jaký internetový protokol používá prohlížeč jako výchozí v případě, že internetová stránka podporuje obě IP a jak rychlá je reakce je jeho reakce, když není stránka na výchozím dostupná (fallback). Na konci testu je konektivita ohodnocena. U hodnocení je možné zobrazit podrobnosti, kde se zobrazí různé typy, jak připojení vylepšit. Po tomto testu bylo například doporučeno získat IPv6 připojení od ISP nebo vyřešit reverzní DNS a podobně. Pro potřeby této práce je služba ideální a to i pro svoje grafické zpracování výstupu výsledků. Byly provedeny čtyři pokusy a výsledky jsou zobrazené na Obrázcích 24 až 27.



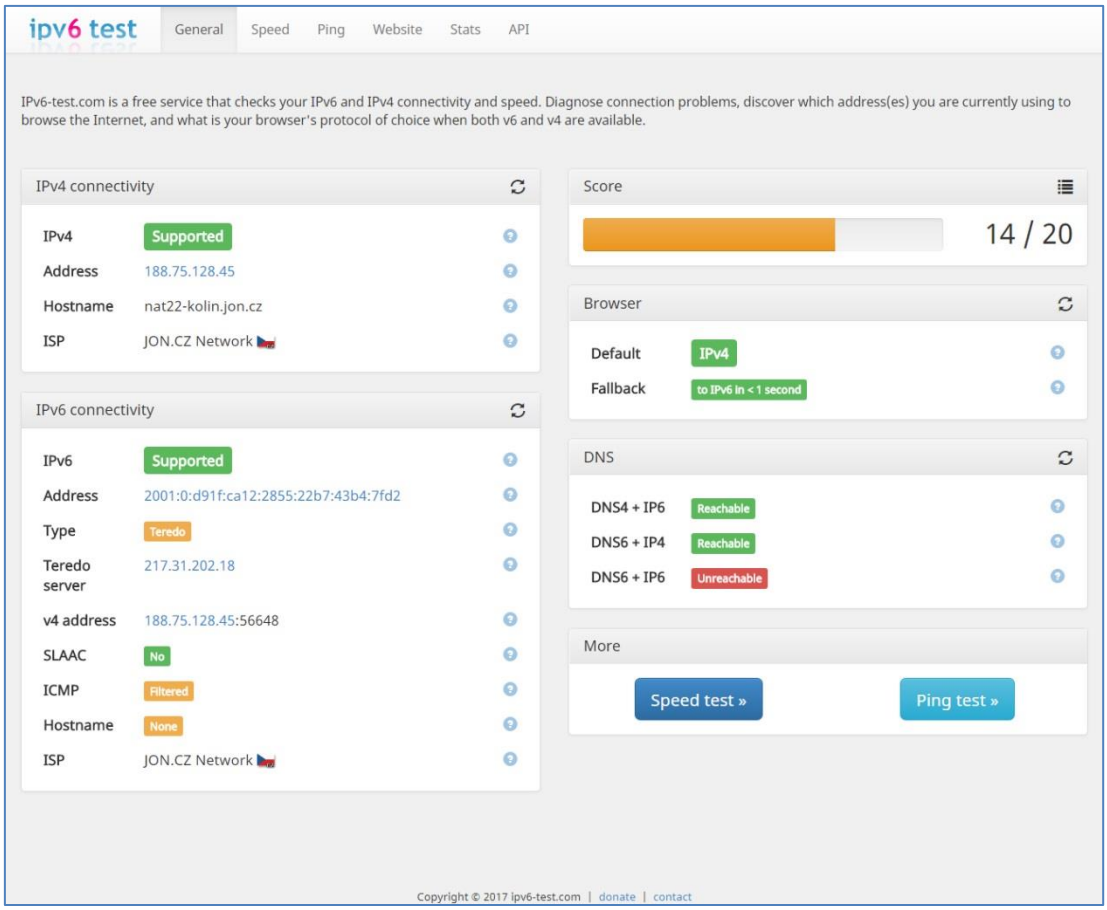
Obrázek 24: IPv6-test.com 1. test



Obrázek 25: IPv6-test.com 2. test



Obrázek 26: IPv6-test.com 3. test

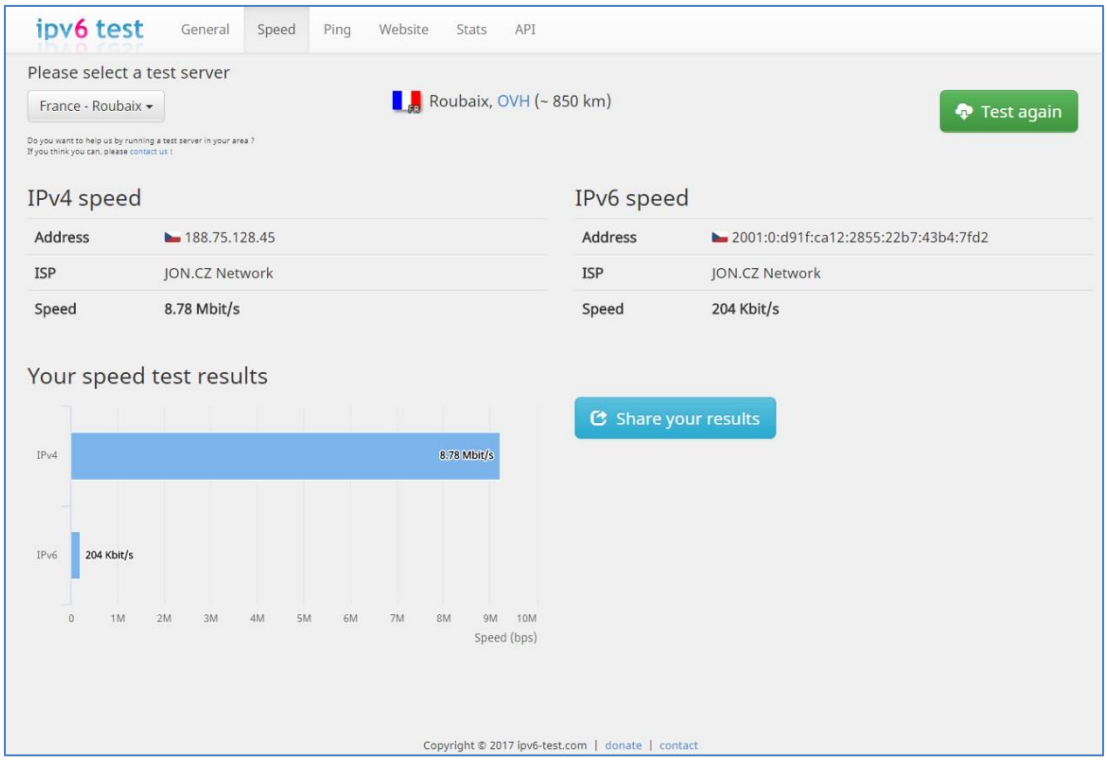


Obrázek 27: IPv6-test.com 4. test

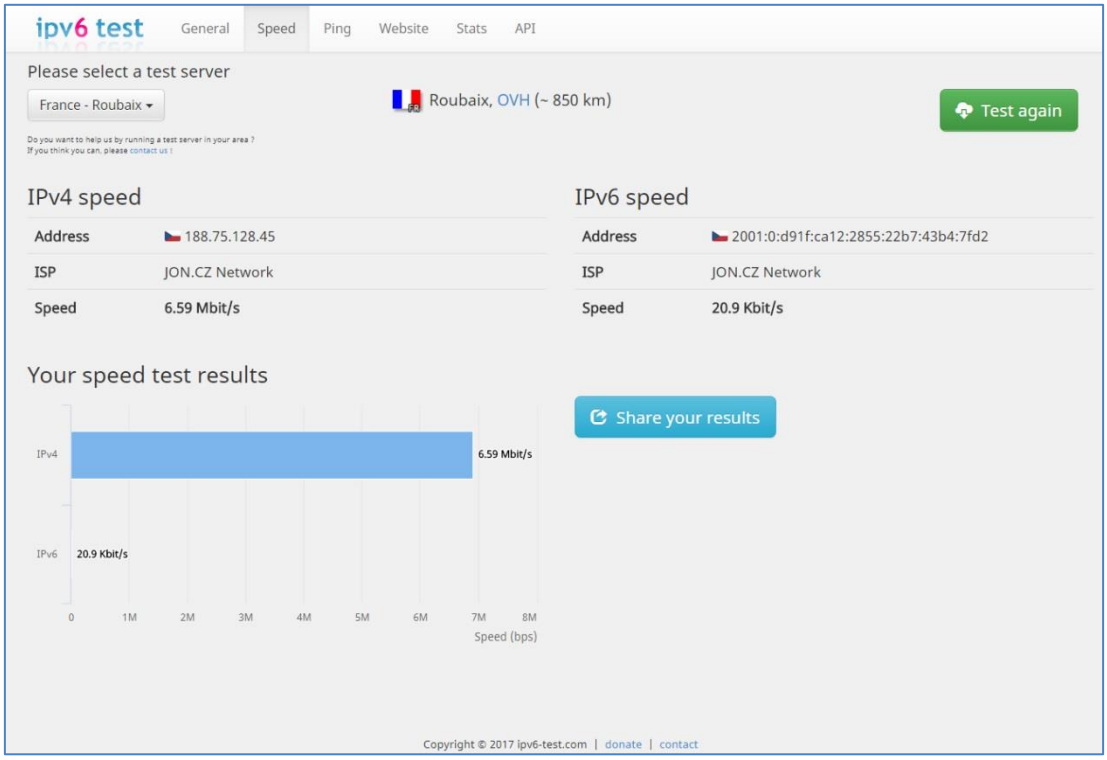
Na první pohled se už projevila nespolehlivost Tereda. Každopádně funguje DNSv4 pro získání IPv6 adresy. V případě DNSv6 je to už horší, testuje se obdržení IPv4 i v6 adresy, ale ani v jednom testu není dostupné obojí. V jednom ze čtyř testů dokonce neuspěl fallback. Nyní je ověřeno, že připojení je funkční pro obě verze internetového protokolu, akorát novější verze s využitím Teredo tunelu má jisté potíže.

6.3.5 Test č. 5

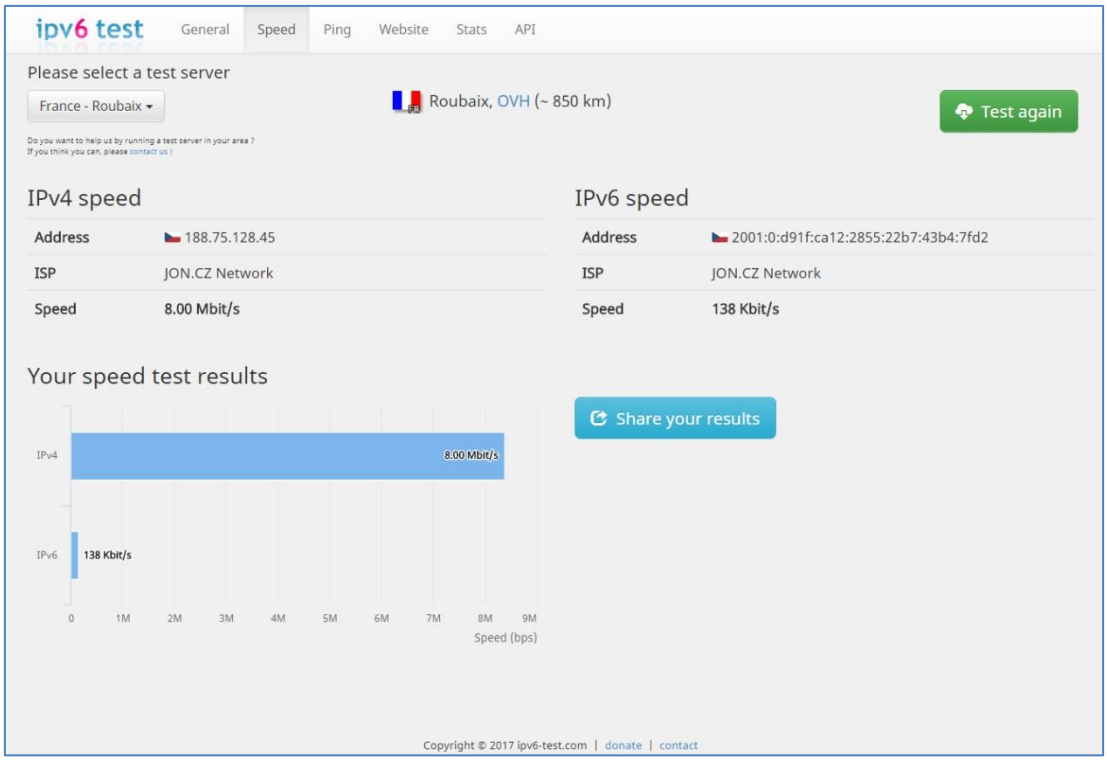
Na řadu přichází test rychlosti připojení. Využita je stejná služba jako v testu č. 4. Na výběr je několik serverů, vůči kterým je možno rychlost otestovat. Na testování jsou vybrány dva servery a u obou jsou provedeny tři pokusy. První server je provozovaný francouzskou společností OVH a nachází se v datacentru ve městě Roubaix. Druhý se nachází v datacentru ve městě Portsmouth a vlastní ho britská společnost ServerHouse. Spuštění je intuitivní, stačí jen přepnout na záložku Speed a vybrat požadovaný server. Na obrázcích 28 až 33 jsou zobrazeny výsledky testování.



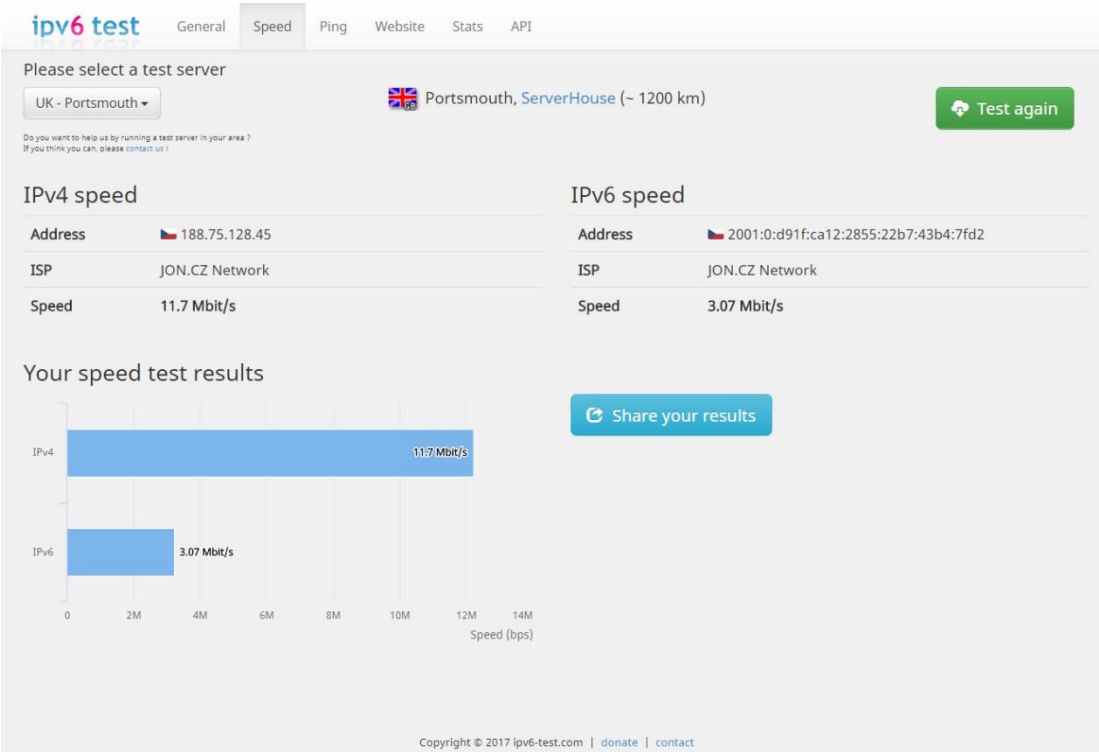
Obrázek 28: IPv6-test.com 1. test rychlosti Roubaix



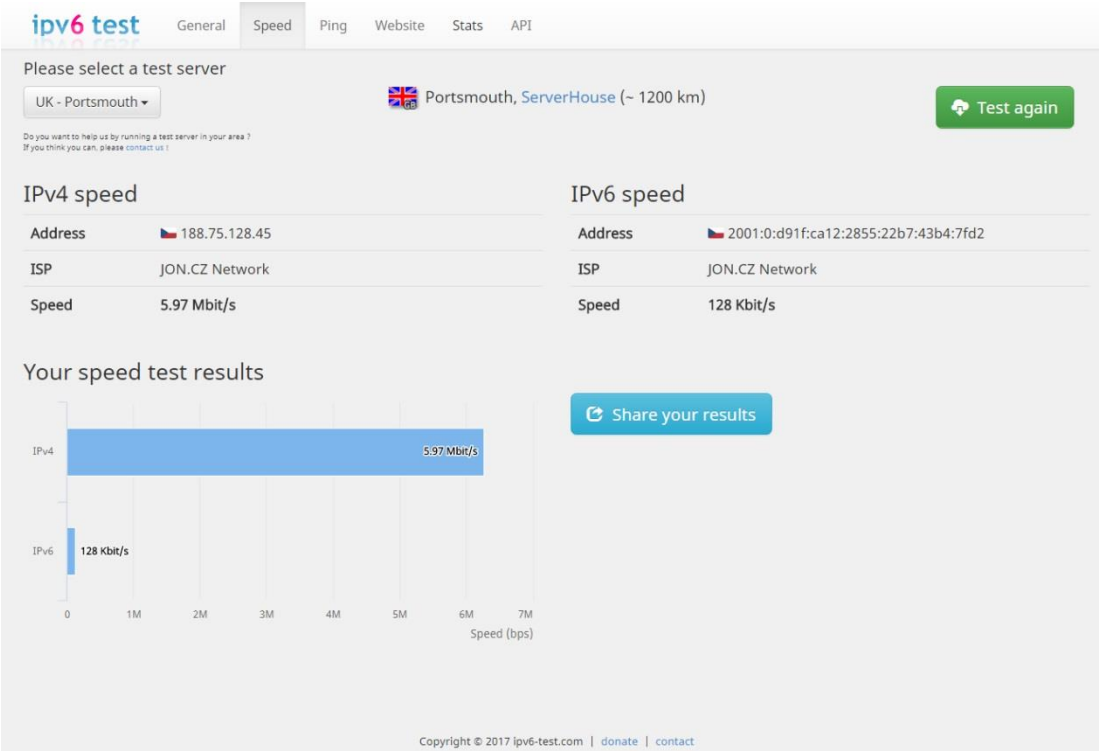
Obrázek 29: IPv6-test.com 2. test rychlosti Roubaix



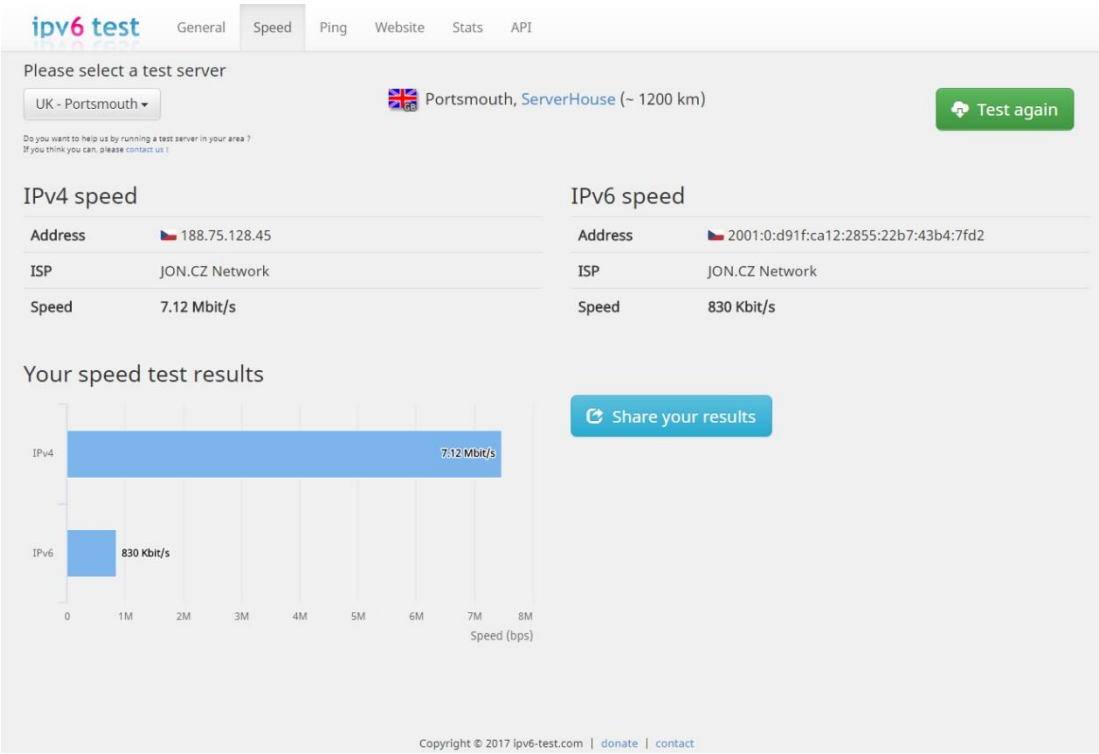
Obrázek 30: IPv6-test.com 3. test rychlosti Roubaix



Obrázek 31: IPv6-test.com 1. test rychlosti Portsmouth



Obrázek 32: IPv6-test.com 2. test rychlosti Portsmouth



Obrázek 33: IPv6-test.com 3. test rychlosti Portsmouth

6.3.6 Test č. 6

Na poslední test byla vybrána opět služná dostupná přes web a to konkrétně *www.test-ipv6.cz*. Princip je podobný jako u přechozí služby, jen výstup není graficky zpracován, ale výsledky testů jednotlivých částí jsou reprezentovány v podobě textového výpisu. Jedná se pouze o test konektivity, tato služba se měřením rychlosti aj. nezabývá. Hlavním důvodem proč jí zvolit je, že v ČR je jejím poskytovatelem sdružení CZ.NIC.

Hned po načtení webové stránky se spustí test automaticky. Po dokončení se byly základní informace o výsledku testu následující:

- IPv4 adresa je 188.75.128.45
- IPv6 adresa je 2001:0:d91f:ca12:2855:22b7:43b4:7fd2
- používá se službu Teredo
- používá se tunelované připojení
- valná většina testovaných IPv6 webových stránek je tunelem dostupná
- internetový prohlížeč upřednostňuje IPv4 před IPv6
- DNS respektive ISP nemá přístup do IPv6

Podrobný výpis výsledů obsahoval následující informace. IPv4 poskytuje JON.CZ s.r.o. a IPv6 sdružení CZ.NIC skrze Teredo. Testy DNS ukázaly, že je možné zobrazit IPv4 i IPv6 webové stránky a to včetně takových, co fungují na obou verzích. Test bez DNS, kdy se přistupuje na webové stránky přes číselnou IP adresu, potvrdil rovněž dostupnost většiny webových stránek. Otestovali se také velké IPv6 pakety a tyto požadavky, včetně DNS požadavků dopadli úspěšně. Jediné co skončilo neúspěchem je, že DNS server poskytovatele internetu není schopný překládat internetové na IPv6 adresy. Celkové hodnocení připojení bylo 9 z 10, což je pozitivní výsledek.

7 Shrnutí výsledků

Konfigurace je jednoduchá, i když jí zřejmě nezvládne každý běžný uživatel. Je potřeba provést sérii příkazů v příkazovém řádku, i když jich je jen pár. Také se musí přidat hodnota DWORD do registru.

Testování nepřineslo dobré výsledky a naopak demonstrují nespolehlivost Tereda. Výsledek prvního testu je velice nepřívětivý. Ze sta odeslaných paketů se jich bylo 35 bez odpovědi, z toho vyplývá úspěšnost pouhých 65%. Odezva taktéž není nijak chvalitebná, v průměru byla 307 ms a dokonce v maximu dosáhla až hodnoty 1704 ms. V přepočtu je to 1,7 vteřiny, což je pro běžný provoz nemyslitelná hodnosta. Pro srovnání stejný test odezvy pro IPv4 dopadl v průměru 6 ms a maximum 14 ms. Druhý a hlavně třetí test jen podporují výsledky prvního.

Potěšující výsledky nepřinesl ani pátý test, kde je pro změnu měřena rychlost připojení pro obě verze IP. Měření vůči francouzskému serveru dopadlo bídně, rychlost IPv6 připojení v průměru byla 121 kbit/s oproti 7,79 Mbit/s u IPv4. S britským serverem to dopadlo lepé. V průměru byly naměřeny hodnoty pro verzi 6 1,3 Mbit a 8,26 Mbit/s pro starší verzi IP. Kromě lepšího výsledku, testování s britským serverem poukázalo znovu na nespolehlivost Tereda, kdy při prvním je naměřeno 3,07 Mbit/s a v dalším je propad na pouhých 128 kbit/s.

Zbývající testy 4 a 6 přinesly příznivější výsledky, které hlavně poukazovaly na to, že ISP neprovozuje DNSv6 a je potřeba se obrátit právě na poskytovatele kvůli nativnímu IPv6. Jinak z pohledu možností využití vypadá Teredo, při správné konfiguraci a vhodně zvoleném serveru, slibně. Bohužel je nespolehlivost a neefektivnost mnohem vážnější.

8 Závěr

V této práci je popsáno, co je to internetový protokol a že má dvě verze IPv4 a IPv6. Obě jsou charakterizovány také. Byly vyzdvihnuty přednosti IPv6. Dále se práce zabývala mechanismy pro přechod mezi oběma protokoly. Popisuje jejich rozdělení a jednotlivé mechanismy. Na závěr je mechanismus Teredo v praktické části podroben sérií testů.

Podle výsledků je evidentní, že se nedá použít na běžný provoz. Server provozovaný výrobcem operačního systému Windows je zřejmě nespolehlivý, pro provedené testování doslova nepoužitelný. Naštěstí v je jeden Teredo server provozován přímo v ČR. Teredo je spíše vhodné pro testování, laboratorní účely a možnost si osahat IPv6 popř. jako záložní varianta.

Kromě všech již zmíněných výhod, má IPv6 další významnou přednost oproti IPv4 týkající se bezpečnosti. Tou je IPSec. Je pravda, že je použitelné i pro verzi 4, ale IPv6 udává, že jeho implementace je povinná. Poskytuje režim autentifikace a šifrování. Autentifikace slouží k ověřování dat, jestli nedošlo cestou k jejich podvržení, změně atd. Šifrování má za úkol to, aby data nepřečetl nikdo jiný než jejich příjemce. IPseec má také 2 režimy. Je-li v transportním režimu, je za IPv6 hlavičku přidána hlavička Authentication Header (AH) a Encapsulating Security Payload (ESP) nebo jen jedna z nich. Upřednostňuje se hlavička ESP, jelikož je obsáhne i to co hlavička AH. Druhý režim je tunelovací. Je-li IPsec takto nastaven, odesílané datagramy zabalí jako data do nového s novými hlavičkami a hlaviček AH/ESP. Toto bývá implementováno často na bránách sítě (směrovačích) mezi, kterými je zabezpečení (IPsec) tunel. Útočník se jen dozví z jaké brány a na jakou bránu byl datagram odeslán. (Kent et al. 2005)

V dnešní době je připojení do IPv6 světa, dost limitované poskytovateli internetu, kteří novou verzi odmítají. Dále i tím, že zde existuje překlad adres (NAT) a nemáme k dispozici veřejnou IPv4 adresu, abychom využili např. tunel od Hurricane Electric. Také tunel brokerů je málo, což situaci neulehčuje. Stejně tak málo IPv6 obsahu na internetu. Já si myslím, že by měli být poskytovatelé internetu vedeni k tomu, aby svým zákazníkům zpřístupnili IPv6 konektivitu. Je pravda, že vybudování infrastruktury IPv6 sítě bude obzvlášť pro ISP finančně nákladné. Pořád tu je minimálně varianta využít mechanismus 6rd, který dovoluje ISP zachovat IPv4 infrastrukturu. Zásah do ní je minimální a hlavní investicí jsou jen relay směrovače mezi 6rd a nativní IPv6.

Většina výhod nové verze už je zmíněna v kapitole o ní. Je tu povinné zabezpečení v podobě IPsec. Zařízení jsou přímo adresovatelná, to také přináší spoustu výhod kupříkladu pro IP telefonii. Je potřeba jít nové verzi naproti. Nabídnout jí uživatelům. Menší ISP mohou zřídit 6rd tunelování pro své zákazníky. Velcí hráči na českém trhu jako například O2, T-Mobile, UPC a další už IPv6 konektivitu nabízí. Počet uživatelů IPv6 sice roste, ale velice pomalu. S jejich nárůstem by došlo i k přibývání obsahu na internetu dostupného přes IPv6.

9 Seznam použité literatury

9.1 Tištěné zdroje

[1] SPORTACK, Mark A., 2004. Směrování v sítích IP. 1. vydání. Brno: Computer Press. ISBN 80-251-0127-4

[2] PARZIALE, Lydia et al., 2006. TCP/IP Tutorial and Technical Overview. 8. vydání. IBM Redbooks. ISBN 0738494682 (vydavatel IBM Redbooks?)

[3] DOSTÁLEK, Libor a Alena KABELOVÁ, 2000. Velký průvodce protokoly TCP/IP a systémem DNS. 2. vydání. Praha: Computer Press. ISBN 80-7226-323-4

[4] KOZIEROK, Charles M., 2005a. The TCP/IP guide: a comprehensive, illustrated internet protocols reference. San Francisco: No Starch Press. ISBN: 1-59327-047-X

[5] SATRAPA, Pavel, 2011. Internetový protokol verze 6. 3. vydání. Praha: CZ.NIC. ISBN 978-80-904248-4-5

[6] HLAVÁČEK, Tomáš. Tunelování v sítích (nejen) IPv6. IT Systems. 2011, 13(6), 60 – 62. ISSN 1802-002X

[7] HAGEN, Sylvia, 2006. IPv6 Essentials. 2. vydání. Sebastopol: O'Reilly Media. ISBN: 0-596-10058-2

9.2 Internetové zdroje

[8] Learn Networking, 2007. How Encapsulation Works Within the TCP/IP Model [online]. cit. 2017-09-07. Dostupné z: http://learn-networking.com/tcp-ip/how-encapsulation-works-within-the-tcpip-model

[9] Information Sciences Institute, University of Southern California, 1981. RFC 791: Internet Protocol [online]. Fremont, California: Internet Engineering Task Force. cit. 2017-09-08. Dostupné z: http://tools.ietf.org/pdf/rfc791.pdf

[10] Institute for Telecommunication Sciences, 2007. Protocol data unit (PDU) [online]. Boulder, Colorado. cit. 2017-09-09. Dostupné z: http://www.its.bldrdoc.gov/fs-1037/dir-028/_4199.htm

[11] USNESENÍ VLÁDY ČESKÉ REPUBLIKY ze dne 8. června 2009 č. 727 ke Zprávě o přechodu na internetový protokol verze 6 (IPv6) [online]. cit. 2017-09-14. Dostupné z: https://kormoran.vlada.cz/usneseni/usneseni_webtest.nsf/0/6BFDE5B071A154C5C12575E5004024F1/$FILE/727%20uv090608.0727.pdf

[12] DEERING, S. et al., 1998. RFC 2460: Internet Protocol, Version 6 (IPv6) Specification [online]. cit. 2017-09-14. Fremont, California: Internet Engineering Task Force. Dostupné z: http://tools.ietf.org/pdf/rfc2460.pdf

[13] NORDMARK, Erik et al., 2005. RFC 4213: Basic Transition Mechanisms for IPv6 Hosts and Routers [online]. Fremont, California: Internet Engineering Task Force. cit. 2017-09-19. Dostupné z: http://tools.ietf.org/pdf/rfc4213.pdf

[14] FILIP, Ondřej, 2011. Tunelovací mechanismy [online]. cit. 2017-09-20. Dostupné z: https://www.nic.cz/files/nic/doc/Computerworld_IPv6_052011.pdf

[15] SATRAPA, Pavel, 2003. IPv6 - přechodové mechanismy (1) [online]. cit. 2017-09-20. Dostupné z: https://www.lupa.cz/clanky/ipv6-prechodove-mechanismy-1/

[16] CONTA, Alex et al., 1998. RFC 2473: Generic Packet Tunneling in IPv6 [online]. Fremont, California: Internet Engineering Task Force. cit. 2017-09-22. Dostupné z: http://tools.ietf.org/pdf/rfc2473.pdf

[17] STEFFANN, Sander et al., 2013. RFC 7059: A Comparison of IPv6-over-IPv4 Tunnel Mechanisms [online]. Fremont, California: Internet Engineering Task Force. cit. 2017-09-24. Dostupné z: http://tools.ietf.org/pdf/rfc7059.pdf

[18] DURAND, Alain et al., 2001. RFC 3053: IPv6 Tunnel Broker [online]. Fremont, California: Internet Engineering Task Force. cit. 2017-10-25. Dostupné z: http://tools.ietf.org/pdf/rfc3053.pdf

[19] SATRAPA, Pavel, 2010. 6rd – nový koncept nasazení IPv6 [online]. cit. 2017-11-02. Dostupné z: https://www.lupa.cz/clanky/6rdnbspndash-novy-koncept-nasazeni-ipv6/

[20] CARPENTER, Brian E. a Keith MOORE, 2001. RFC 3056: Connection of IPv6 Domains via IPv4 Clouds [online]. Fremont, California: Internet Engineering Task Force. cit. 2017-11-02. Dostupné z: http://tools.ietf.org/pdf/rfc3056.pdf

[21] SAVOLA, Pekka et al., 2004. RFC 3964: Security Considerations for 6to4 [online]. Fremont, California: Internet Engineering Task Force. cit. 2017-11-02. Dostupné z: http://tools.ietf.org/pdf/rfc3964.pdf

[22] HUITEMA, Christian a Microsoft Corporation, 2001. RFC 3068: An Anycast Prefix for 6to4 Relay Routers [online]. Fremont, California: Internet Engineering Task Force. cit. 2017-11-02. Dostupné z: http://tools.ietf.org/pdf/rfc3068.pdf

[23] DESPRES, Remi a RD-IPtech, 2010. RFC 5569: IPv6 Rapid Deployment on IPv4 Infrastructures (6rd) [online]. Fremont, California: Internet Engineering Task Force. cit. 2017-11-06. Dostupné z: http://tools.ietf.org/pdf/rfc5569.pdf

[24] TEMPLIN, Fred L. et al., 2008. RFC 5214: Intra-Site Automatic Tunnel Addressing Protocol (ISATAP) [online]. Fremont, California: Internet Engineering Task Force. cit. 2017-11-09. Dostupné z: http://tools.ietf.org/pdf/rfc5214.pdf

[25] PODERMAŃSKI, Tomáš a Matěj GRÉRG, 2011. IPv6 Mýty a skutečnost, díl VIII. - Přechodové mechanizmy [online]. cit. 2017-11-09. Dostupné z: https://www.lupa.cz/clanky/ipv6-myty-a-skutecnost-dil-viii-prechodove-mechanizmy/

[26] CARPENTER, Brian E. et al., 1999. RFC 2529: Transmission of IPv6 over IPv4 Domains without Explicit Tunnels [online]. Fremont, California: Internet Engineering Task Force. cit. 2017-11-11. Dostupné z: http://tools.ietf.org/pdf/rfc2529.pdf

[27] DURAND, Alain et al., 2011. RFC 6333: Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion [online]. Fremont, California: Internet Engineering Task Force. cit. 2017-11-13. Dostupné z: http://tools.ietf.org/pdf/rfc6333.pdf

[28] LI, Xing et al., 2011. RFC 6145: IP/ICMP Translation Algorithm Exhaustion [online]. Fremont, California: Internet Engineering Task Force. cit. 2017-11-18. Dostupné z: http://tools.ietf.org/pdf/rfc6145.pdf

[29] IPv6.cz, 2011. Překlad IP a ICMP [online]. cit. 2017-11-18. Dostupné z: https://www.ipv6.cz/P%C5%99eklad_IP_a_ICMP

[30] BAGNULO, Marcelo et al., 2011. RFC 6146: Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers [online]. Fremont, California: Internet Engineering Task Force. cit. 2017-11-20. Dostupné z: http://tools.ietf.org/pdf/rfc6146.pdf

[31] HUANG, Bill et al., 2012. RFC 6535: Dual-Stack Hosts Using "Bump-in-the-Host" (BIH) [online]. Fremont, California: Internet Engineering Task Force. cit. 2017-11-22. Dostupné z: http://tools.ietf.org/pdf/rfc6535.pdf

[32] HUITEMA, Christian a Microsoft Corporation, 2006. RFC 4380: Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs) [online]. Fremont, California: Internet Engineering Task Force. cit. 2017-11-26. Dostupné z: http://tools.ietf.org/pdf/rfc4380.pdf

[33] THALER, Dave et al., 2010. RFC 5991: Teredo Security Updates [online]. Fremont, California: Internet Engineering Task Force. cit. 2017-11-26. Dostupné z: http://tools.ietf.org/pdf/rfc5991.pdf

[34] KENT, Stephen et al., 2005. RFC 4301: Security Architecture for the Internet Protocol [online]. Fremont, California: Internet Engineering Task Force. cit. 2017-12-01. Dostupné z: http://tools.ietf.org/pdf/rfc4301.pdf

Univerzita Hradec Králové
Fakulta informatiky a managementu
Akademický rok: 2017/2018

Studijní program: Aplikovaná informatika
Forma: Prezenční
Obor/komb.: Aplikovaná informatika (ai3-p)

Podklad pro zadání BAKALÁŘSKÉ práce studenta

PŘEDKLÁDÁ:	ADRESA	OSOBNÍ ČÍSLO
Davídek Martin	Morávkova 1160, Kolín - Kolín V	I1201464

TÉMA ČESKY:

Mechanismy přechodu mezi IPv4 a IPv6

TÉMA ANGLICKY:

Mechanisms of migration between IPv4 and IPv6

VEDOUCÍ PRÁCE:

Ing. Pavel Blažek - KIT

ZÁSADY PRO VYPRACOVÁNÍ:

Cílem této bakalářské práce je prozkoumat internetové protokoly verze 4 a verze 6 a mechanismy přechodu mezi oběma verzemi protokolu. Na závěr je praktická část, ve které bude některý mechanismus aplikován.

1. Úvod
2. Teoretická část
2.1. Internetové protokoly
2.2. Mechanismy
3. Praktická část
4. Závěr
5. Použité zdroje

SEZNAM DOPORUČENÉ LITERATURY:

1. DOSTÁLEK, Libor a Alena KABELOVÁ, 2000. Velký průvodce protokoly TCP/IP a systémem DNS
2. HAGEN, Sylvia, 2006. IPv6 Essentials
3. SATRAPA, Pavel, 2011. Internetový protokol verze 6
4. RFC dokumenty www.ietf.org

Podpis studenta: ..... **Datum:** 14. 12. 2017

Podpis vedoucího práce: ..... **Datum:** 14. 12. 2017