IPv6 Mýty a skutečnost, díl VIII. - Přechodové mechanizmy

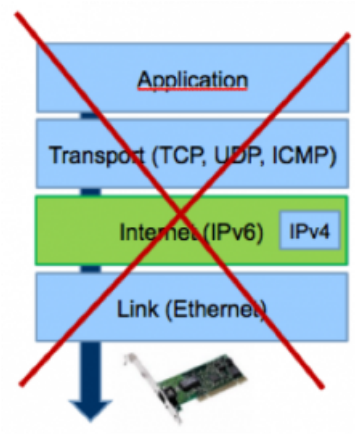
31. 3. 2011 6:25 Tomáš Podermański, Matěj Grégr

Seriál Pohněme s IPv6

- IPv6 Mýty a skutečnost, díl V. - Zjednodušené hlavičky
- IPv6 Mýty a skutečnost, díl VI. - Bezpečnostní mechanizmy
- IPv6 Mýty a skutečnost, díl VII. - Podpora Multicast a anycast provozu
- IPv6 Mýty a skutečnost, díl VIII. - Přechodové mechanizmy
- IPv6 Mýty a skutečnost, díl IX. - Quo Vadis, IPv6?

Všechny díly seriálu

Protokol IPv6 byl již od počátku vytvářen jako protokol nekompatibilní s dříve používaným protokolem IPv4. To na jednu stranu umožnilo tvůrcům neomezovat se při návrhu vazbou na původní protokol, ale na druhou stranu komplikuje zavádění IPv6 v sítích. Na možnosti koexistence obou protokolů se zaměříme v dnešním dílu.



IPv6 není rošířením IPv4

Často se můžeme setkat s představou, že protokol IPv6 je jakousi nadmnožinou protokolu IPv4 a v nových instalacích stačí zavádět pouze IPv6. Tato představa je naprosto mylná, protože, jak jsme si již řekli dříve, protokoly IPv4 a IPv6 jsou vzájemně nekompatibilní. Stejně tak je na omylu ten, kdo si myslí, že se jej problematika IPv6 netýká, protože má dostatek IPv4 adres, anebo že mu IPv6 vyřeší akutní problém nedostatku IPv4 adres. Základní koncept předpokládá, že všichni a všechno bude časem dostupné po IPv6 a až pak je možné prohlásit, že IPv6 řeší problém nedostatku adres jednou pro vždy. V praxi to tedy znamená, že pokud chceme komunikovat s oběma světy, je nezbytná podpora obou protokolů na všech úrovních, tj. od aplikace přes operační systém, síťovou infrastrukturu, směrování, připojení k ISP až po klientské systémy. Již na první pohled je zřejmé, že doba, po kterou bude realizován přechod od IPv4 do IPv6, nebude zrovna krátká. Ponechme stranou to, jak náročné, omezující a zda vůbec možné by bylo vytvořit protokol kompatibilní, a podívejme se na možnosti koexistence protokolu IPv4 a IPv6, a zejména na prostředky usnadňující jejich společné použití.

Technologickým garantem seriálu Pohněme s IPv6 je CZ.NIC .

Celkově lze tyto prostředky, které označujeme jako přechodové mechanizmy (*IPv6 transition mechanisms*), rozdělit do tří základních skupin.

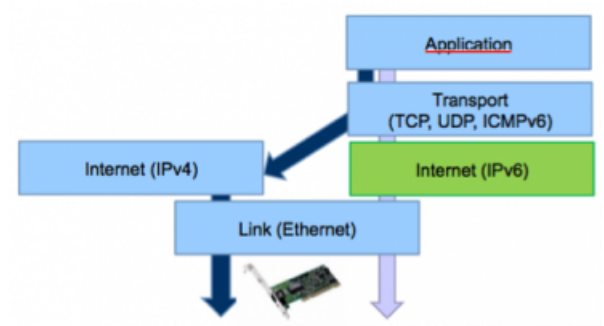
- **Současné provozování obou protokolů** v systémech. Umožňuje současný chod aplikacím využívající jak IPv4, tak IPv6 protokol.
- **Tunelovací mechanizmy** propojující jednotlivé IPv6 sítě s využitím tranzitu po IPv4.
- **Překladové mechanizmy** zajišťující konverzi komunikace mezi IPv6 a IPv4.

V průběhu procesu standardizace vznikla celá řada specifikací a návrhů, které si postupem času buď našly své místo v existujících systémech, anebo byly zhodnoceny jako slepé uličky vývoje a zrušeny. Přechodových mechanizmů je celá řada a pravděpodobně budou vznikat ještě další. Zkusme se seznámit s těmi nejčastějšími a případně upozornit na úskalí jejich používaní.

Dualstack a mapované adresy

První místo, kde narazíme na rozdílnost protokolů, je aplikace. Chceme-li aplikaci provozovat jak pod IPv4, tak pod IPv6, musí aplikace umět pracovat s každým protokolem samostatně. Mechanizmu podpory obou protokolů říkáme dvojí zásobník neboli dualstack.

Vzhledem k tomu, že úprava se týká úplně všech aplikací využívajících aplikační rozhraní pro práci s protokolem TCP/IP (IP socket), vznikla snaha, jak co nejvíce zjednodušit jejich úpravu a současně umožnit nově vznikajícím aplikacím již automaticky využívat oba protokoly s minimálním úsilím pro tvůrce aplikace. Mechanizmu, který umožňuje jednotně využívat oba protokoly, říkáme IPv4 mapované IPv6 adresy (IPv4-mapped IPv6 addresses).



IPv4 mapované adresy do IPv6

Základní myšlenkou mapování adres je vytvoření aplikačního rozhraní (API), které se co nejvíce podobá stávajícímu API využívanému v protokolu IPv4. Aplikace (například web server) pak pracuje s tímto rozhraním a nemusí řešit, zda na nižší vrstvě byla data přijata IPv4 nebo IPv6 protokolem. Aplikace s rozhraním pracuje vždy jako s IPv6 s tím, že pokud chce využívat protokolu IPv4, využije speciální adresový prostor `::ffff:<IPv4 adresa>`. Volba správného protokolu je pak už záležitostí nižších vrstev a aplikace se tímto nemusí trápit.V praxi se však tato myšlenka příliš neujala a některé systémy mapované adresy vůbec nepodporují (Windows) a nebo se je snaží potlačit z bezpečnostních důvodů. Většina aplikací tedy musí mít implementovanou plnohodnotnou podporu jak IPv4, tak IPv6 protokolu nezávislým kódem.

IPv6 – dej přednost v jízdě

Společně s dual-stackem úzce souvisí problematika překladu doménových jmen na IP adresy. Pro převod jmen na adresy se v IPv6 používají AAAA záznamy. Jedná se o ekvivalenty A záznamů určených pro převod na IPv4 adresy. V případě, že se klient snaží přistupovat na službu inzerovanou pod jménem, například www.datoveschranky.info, dotáže se klientský systém rekurzivního DNS serveru nejdříve na AAAA záznam odpovídající tomuto jménu. V případě, že získá negativní odpověď, pokusí se získat odpovídající A záznam. Tímto je zaručeno, že klientský systém bude přednostně využívat připojení s využitím protokolu IPv6.

Zde nastává první skupina problémů. Samotná IPv6 konektivita je mnohdy provozována zatím v experimentálním režimu, a ne vždy funguje zcela spolehlivě. Někdy dokonce nefunguje vůbec v důsledku šíření špatných autokonfiguračních informací službou Internet Connectin Sharing v systémech Windows Vista/7 (viz. díl věnovaný autokonfiguraci). Další problém souvisí se způsobem vytváření připojení k serveru. V případě, že na serveru je služba spuštěna pouze pro IPv4 a pro IPv6 je spojení odmítnuto, řada klientů se již nepokouší navázat spojení prostřednictvím IPv4. Lze bezpochyby namítnout,

že služby, které nejsou dostupné po IPv6, nemají být inzerovány v DNS. V praxi je však na jednom serveru zpravidla provozováno více služeb a vše je inzerováno pod jedním jménem nebo s využitím CNAME záznamů. Při zavádění IPv6 je tedy třeba myslet na to, aby všechny služby, které jsou v DNS inzerovány pod společným jménem, podporovaly rovněž protokol IPv6.

Z praktického hlediska by se mnohdy hodilo, aby A záznamy byly upřednostňovány před AAAA. Ty by se použily pouze v případě, že není k dispozici odpovídající A záznam. Tímto bychom si vynutili přednostní použití protokolu IPv4. Bohužel upřednostňování příslušného typu záznamů je ovlivnitelné pouze tvůrcem aplikace a mnohdy je chování „zadrátováno“ přímo v jejím kódu. Částečně lze toto chování ovlivnit nastavením tabulky preferencí na úrovni operačního systému (RFC3484). To však nemusí jednotlivé aplikace respektovat. Občas je možné tuto volbu konfigurovat, například parametrem při spuštění – vesměs u řádkových příkazů.

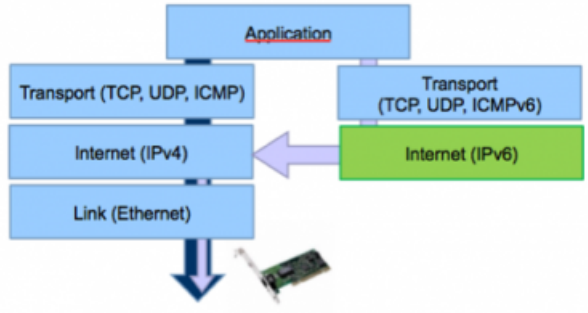
Důsledkem uvedeného chování je, že aplikace se snaží problém řešit po svém, chování jednotlivých aplikací se tím pádem liší a poměrně špatně se předem odhaduje. Některé aplikace se pokusí navázat spojení prostřednictvím IPv6 a v případě neúspěchu se již nesnaží navázat spojení po IPv4 (například všechny aplikace na iOS). Další typ aplikací postupně zkouší všechny adresy získané jak z AAAA, tak z A záznamů, dokud se nepodaří navázat spojení. Velkou nepříjemností je, že z hlediska uživatele je zpravidla proces výběru adres naprosto neovlivnitelný. Například v žádném z dnes používaných webových prohlížečů není možné, byť jen dočasně, vynutit si upřednostňování IPv4 před IPv6. Běžného uživatele pak dokáže pěkně popudit, když každé načtení stránky, například katastru nemovitostí, anebo datových schránek, je doprovázeno čekáním v řádu desítek vteřin. To jen proto, že ten den není IPv6 zrovna v „nejlepší kondici“, anebo sousedovo PC začalo nekontrolovaně šířit nevalidní autokonfigurační informace do naší sítě. S protokolem IPv4 (tedy bez IPv6) by uživateli fungovalo vše dle očekávání, což staví protokol IPv6 do velice negativního světla a spíš vede k tomu, že se uživatelé budou snažit IPv6 na svých systémech všemožně deaktivovat.

Tunelovací mechanizmy

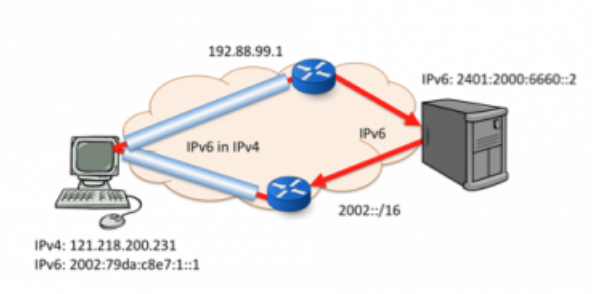
Tunelování je obecně zabalení jednoho protokolu do druhého. IPv6 paket je zapouzdřen do IPv4 a odeslán přes IPv4 síť. Pokud směrovač na druhé straně tunelu obdrží takto zapouzdřený paket, odstraní z něj IPv4 hlavičku a získaný původní IPv6 paket pošle klasicky podle svých směrovacích tabulek. V tomto kontextu si tunelování obecně můžeme představit jako propojování IPv6 ostrůvků do globální IPv6 sítě s využitím IPv4 infrastruktury.

6to4

Tunelovací protokol 6to4 je asi tím nejpoužívanějším tunelovacím protokolem pro zajištění IPv6 konektivity a propojení IPv4 sítí (RFC 3056). Ke svému fungování potřebuje alespoň jednu veřejnou IPv4 adresu. Pomocí ní si vytvoří IPv6 adresu, kde za prefix `2002::/16` přidá dalších 32 bitů své IPv4 adresy. Vznikne tak prefix `/48`. Dalších 16 bitů je určeno pro podsítě, které si může správce nastavit dle libosti. Zbývajících 64 bitů je určeno pro okolní uzly v síti, které si vytvoří v takovém prefixu, s využitím prostředků autokonfigurace, vlastní IPv6 adresu. Při odeslání paketu zabalí 6to4 tunel IPv6 paket do IPv4 s číslem protokolu 41. Tento paket je odeslán na nejbližší dostupný 6to4 relay směrovač – tedy směrovač, který má k dispozici nativní IPv4 a IPv6 připojení a dokáže provést rozbalení (dekapsulaci) paketu. Pro tyto 6to4 relay směrovače je vyhrazena anycastová adresa 192.88.99.1. Při zpětné komunikaci je opět celý prefix `2002::/16` směrován na nejbližší 6to4 relay směrovač, a ten zajistí zabalení (enkapsulaci) IPv6 paketu a odeslání do IPv4 infrastruktury. Princip protokolu ilustruje obrázek 3.3.1.



Architektura protokolu 6to4



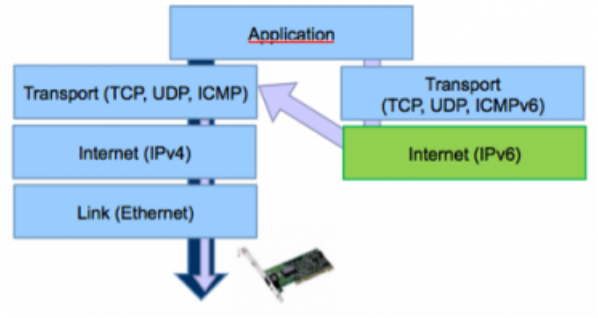
Tunelovací protokol 6to4, převzato z http://www.potaroo.net/

Mezi klíčové problémy mechanizmu 6to4 patří to, že pracuje asymetricky. Paket se může vracet jinou cestou, než byl odeslán. Tím se značně komplikuje jakákoliv diagnostika. Velké množství spojení bývá přerušeno, například díky firewallu, který je často konfigurován tak, že má zahazovat neznámé IP protokoly, tedy i protokol 41. Výsledkem je, že velká část spojení u 6to4 tunelů je velice nespolehlivá. Naštěstí jsou systémy ve výchozím stavu zpravidla konfigurovány tak, že pro přístup na služby inzerované jak v IPv4, tak v IPv6 (tedy prostřednictvím jak A, tak AAAA záznamu) upřednostňuje protokol IPv4.

Poněkud jiná situace už ovšem nastává v případě, že se takový prefix začne šířit službou Internet Connection Sharing (viz. díl věnovaný autokonfiguraci). Některé implementace OS jej už považují za nativní IPv6 konektivitu (MAC OS < 10.6.5, Android < 2.2), a tedy i upřednostňují pro přístup ke službám protokolem IPv6 a veškerá komunikace je vedená zařízením, které takovýto prefix šíří (většinou nevědomě).

Teredo

Tunelovací protokol Teredo primárně slouží jako tranzitní prostředek pro zařízení bez veřejné IPv4 adresy, tedy zařízení umístěná za NATem. Díky tomu, že ke komunikaci používá protokol UDP, dokáže se s absencí veřejné adresy docela dobře vypořádat. Počáteční inicializace musí proběhnout ze strany klienta. Poté, co je ustanoveno spojení, získá klient využívající Teredo protokol veřejně dostupnou IPv6 adresu.



Architektura protokolu Teredo

Díky tomu, že jsou IPv6 datagramy přenášené přes UDP často na dynamických portech, správce většinou netuší, že tento provoz danou sítí prochází. Na hraničním firewallu nelze filtrovat UDP, pokud by se každý paket nerozbaloval a nekontroloval přítomnost zapouzdření Teredo. Jeden z hlavních bezpečnostních problému spojeným s Teredem je možnost šíření virů a malware. Velké množství domácností dnes používá domácí směrovač s NATem jako přístupový bod k internetu pro několik zařízení. Byť NAT nelze považovat za obecný bezpečnostní prvek, jistou formu ochrany poskytuje. Takto všechna zařízení „skrytá“ za NATem jsou najednou přímo dostupná po veřejné IPv6 adrese, a tedy mnohem lépe dostupná pro napadení. To vše se na platformě MS Windows děje zcela automaticky, aniž by uživatel o něčem věděl.

Obecně lze Teredo považovat spíše jako krajně nouzové řešení. Spolehlivost a odezvy Teredo tunelů jsou zpravidla velice špatné. Implementací pro systémy, které nejsou z produkce Microsoft, je Miredo . Detailnější popis Tereda můžete najít v článku PAVLA SATRAPY na Lupě .

ISATAP

Mechanizmus ISATAP (*Intra-Site Automatic Tunnel Addressing Protocol*) je navržen zejména pro firemní prostředí. V síti se zřídí jeden nebo více ISATAP směrovačů, které praktický slouží jako zakončení tunelu. Klienti se o existenci ISATAP serverů dozví zpravidla prostřednictvím DNS, kde je požadován alespoň

jeden záznam *isatap. doména.* Pokud záznam v DNS uveden není, mechanizmus ISATAPu se nepoužije. Tunelování se provádí podobně jako u 6to4 pomocí zapouzdření v protokolu 41. IPv4 adresa je zakódována ve spodních bitech adresy IPv6, což umožňuje implementovat ISATAP směrovač jako bezstavové zařízení.

Tunel Broker

Dalším možným řešením zavedení IPv6 je využití některé z veřejné služby poskytující IPv6 konektivitu. V takovém případě je z vašeho PC, nebo ze zařízení, které připojuje síť, vytvořen tunelovaný propoj s předem dohodnutým koncovým bodem. Poskytovatelem služby Tunel Broker je zpravidla přidělen celý prefix o délce 48 bitů, a je tedy možné připojit celé sítě. Tuto možnost využijete v případě, že chcete začít experimentovat ve vaší síti se zaváděním nativního připojení IPv6 a váš poskytovatel zatím IPv6 nepodporuje. Příkladem provozovatele služby Tunel Broker může být Hurricane Electric , anebo SixXS . Služby tunelovaného připojení IPv6 jsou poskytovány bezplatně. Hurricane Eletric používá pro tunelování IP číslo protokolu 41 stejně jako 6to4 a ISATAP. SixXS využívá technologii tunelování AIYIA (Anything in Anything), pro kterou je potřeba klientský software (k dispozici zdarma pro všechny platformy). Určitou nevýhodou může být, že pro získání IPv6 prefixu u SixXS musíte napsat krátkou anglickou esej, proč chcete prefix získat. Hurricane Eletric nic takového nevyžaduje, ale podle subjektivních zkušeností je rychlost připojení horší.

Bezpečnost tunelů

Obsah tunelů je skrytý před správcem sítě a v dnešní době neexistuje efektivní filtrování provozu procházejícího uvnitř tunelovacího protokolu. Zejména u automaticky vytvářených tunelů (6to4, Teredo) může útočník snadno obejít nastavenou bezpečnostní politiku, kterou implementuje většinou hraniční firewall. Uživatel o takto vytvořených tunelech většinou netuší, protože jsou automaticky aktivovány po startu systému. Novější systémy Windows se snaží získat IPv6 konektivitu všemi dostupnými prostředky, a pokud není k dispozici nativní IPv6 připojení, jeden ze způsobu tunelování většinou uspěje.

Další bezpečnostní problém vytváří tunelovací mechanizmy používající IP protokol 41. Tyto mechanizmy lze zneužít pro vytváření směrovacích smyček, a tedy k DDoS útoku. Tunelovací směrovače totiž automaticky předpokládají, že cílová adresa příchozího IPv6 paketu je vždy správná adresa uzlu, který je přes tunel dostupný. Útočník tak může vytvořit upravený paket, který je směrován tunelovacím provozem na uzel, který se tunelování neúčastní. Ten pak může paket přeposlat nativním rozhraním do IPv6 sítě. Paket je pak směrován zpět na vstup, odkud je paket směrován opět do tunelu. Smyčka je ukončena pouze při dosažení nulové hodnoty v políčku *Hop Limit* v hlavičce IPv6 datagramu .

Automatické tunelovací techniky lze asi nejlépe vystihnout úslovím „víc škody než užitku“. Ve světle nastíněných problémů vznikají aktivity s cílem prohlásit alespoň některé z těchto technik za historický artefakt. Je docela možné, že v době čtení tohoto článku již v tomto směru byly podniknuty některé kroky v rámci IETF skupiny v6ops na 80. jednání v Praze (viz. agenda v6ops pro 31. března 2011).

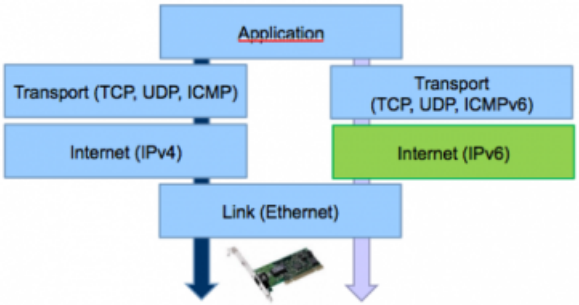
K čemu se tunely ve skutečnosti používají?

Poměrně zajímavým problémem je zkoumání vlastního obsahu tunelu. Většina současných nástrojů pro monitoring sítí neumožňuje zkoumat obsah tunelované komunikace, která je tímto zahalena tajemnou mlhou. Jedním z pokusů o její poodhalení je aktivita vývojové skupiny programovatelného hardware na půdě sdružení CESNET. Do monitorovacích karet určených pro sběr NetFlow statistik se podařilo implementovat rozšíření umožňující analyzovat obsah všech 3 základních tunelovacích protokolů. Řešení bylo po určité období nasazeno v páteřní síti CESNET II. Závěry jsou poměrně zneklidňující. Tunelovaný provoz zatím výrazně převažuje nad nativním. Z hlediska počtu toků jednoznačně převažuje tunelovací mechanizmus Teredo. Nejpodstatnější ovšem je, že naprostá většina datového provozu přísluší provozu různým p2p sítím a blíže neidentifikovatelnému provozu. Bližší informace je možné získat v prezentaci *Tunneled IPv6 Traffic Monitoring using NetFlow* .

Duální provozování IPv4 a IPv6 sítě

Mezi nejpřirozenější způsoby přechodu na IPv6 je paralelní provozování IPv4 a IPv6 infrastruktury. Tento

postup nelze ani nazvat přechodovým mechanizmem, protože je to věc, která funguje v síti naprosto přirozeně bez potřeby vytváření dalších prostředků. Výhodou tohoto řešení je, že oba protokoly jsou v koncových sítích zavedeny nativně a je na volbě koncového uzlu, který protokol bude využívat. Pro označení tohoto režimu sítě se používá trošku nepřesný termín Dual Stack Network.



Plnohodnotný dualstack uzel

Ze všech uvedených mechanizmů představuje nejnáročnější variantu pro provoz sítě. Jak pro protokol IPv4, tak IPv6 musíte provozovat dvě logické sítě (byť na jedné fyzické infrastruktuře) a tedy konfigurovat vše 2× – přidělování adres, směrovací protokoly, DNS, pravidla firewallu, ochranné prvky (DHCP snooping atd.), QoS atd. V každém případě se jedná prakticky o nejspolehlivější cestu pro současné připojení do IPv4 i IPv6 sítě. Současně je tento způsob považován za „nejčistší“ formu zavádění IPv6 v sítích.

V případě dual stack sítí je protokol IPv4 provozován na privátních IPv4 adresách a IPv6 na globálních, případně ULA adresách (viz. díl věnovaný podpoře end-to-end služeb).

Je nutno počítat s tím, že provozování sítí v tomto režimu bude trvat poměrně dlouhou dobu, a že dnes se jedná v podstatě o jediný spolehlivě fungující způsob zavádění IPv6 v koncových sítích. Pokud je to jen trochu možné a neexistují technologické překážky, je zcela jistě nejlepší upřednostnit tento způsob zavádění IPv6 v síti.

Překladové mechanizmy – NAT64, DNS64

Jak jsme si naznačili výše, pro mnohé sítě by byla jistě velice lákavá myšlenka minimalizovat náklady na provoz sítě tím, že v síti budeme provozovat pouze IPv6 protokol. Veškerá komunikace uvnitř sítě by se odehrávala s využitím tohoto protokolu a případný přechod do světa IPv4 by zajistilo jedno zařízení na lince směrem k poskytovateli, nebo až rovnou poskytovatel. Výhody jsou zřejmé, zjednodušená konfigurace bezpečnostní politiky, snazší údržba síťových zařízení, služby provozované pouze pod protokolem IPv6 atd.

Technika překladu protokolu v podobě NAT64 vychází z původního návrhu v podobě překladu protokolu pojmenovaného jako NAT-PT (*NAT Protocol Transition*). Tento návrh předpokládal, že bude možné realizovat překlad protokolu z IPv6 do IPv4 a zpět. Původní myšlenka přeložit cokoliv na cokoliv však narazila na řadu implementačních problémů a návrh byl v roce 2007 prostřednictvím RFC 4966 prohlášen za mrtvý. Mechanizmus však nezmizel bez náhrady. Výrazně okleštěná podoba původního NAT-PT přetrvala v podobě NAT64 a DNS64. Uvedené mechanizmy předpokládají, že zařízení bude realizovat překlad IPv6 adres na jednu nebo více IPv4 adres. Mechanizmus se velice podobá klasickému NATu (resp. NAPTu) z protokolu IPv4 s tím, že na straně privátní sítě je použit IPv6 protokol.

Na rozdíl od běžného NATu (NAPTu) není možné překlad realizovat pouhým překladem adres a portů, ale je nutné být vybaven dalším doprovodným mechanizmem v podobě DNS64. Úkolem DNS64 je vytvořit klientovi v koncové síti iluzi, že server, na který se snaží přistupovat, je dostupný IPv6 protokolem a odpovídajícím způsobem mu „podstrčit“ AAAA adresu. Klient pak zahájí komunikaci na cílovou adresu uvedenou v AAAA záznamu a v případě, že se cíl nachází v IPv4 světě, mechanizmus NAT64 zajistí konverzi takového paketu do IPv4 protokolu. Při předání informace do IPv4 si poznačí stavovou informaci a při obdržení odpovědi ze serveru provede konverzi opačným postupem.

Z pohledu budování sítí jsou úvahy o praktickém nasazení mechanizmu NAT64, DNS64 zatím hodně předčasné. Důvod je velice prostý. Do doby než budou vyřešeny problémy na úrovni lokální sítě, zejména mechanizmus autokonfigurace, je diskuse o těchto mechanizmech naprosto bezpředmětná. V případě nasazování NAT64, DNS64 musí všechna zařízení podporovat protokol IPv6. Pokud musíme v síti provozovat protokol IPv4, byť jen na několika zařízeních, ztrácí nasazení této technologie téměř smysl.

V takovém případě je snazší provozovat v síti oba protokoly.

Nasazení NAT64, DNS64 bude mít patrně velký význam v případě, že IPv6 bude nabízet dostatečně spolehlivé prostředí pro provoz sítě, a množství přenášených dat po IPv6 bude výrazně převažovat nad protokolem IPv4. Pokud se však Internet dostane do tohoto stavu, je velice pravděpodobné, že prakticky všechny významné služby již budou dostupné po IPv6, a použití IPv4 v globálním Internetu ztratí velice rychle smysl. Toto je ovšem hudba hodně daleké budoucnosti – pokud vůbec nastane. Cokoliv předvídat v tomto směru je prakticky nemožné.

Závěr

Přechodové mechanizmy představují důležitý prvek implementace IPv6 při nasazování protokolu do existujících sítí. Nejpřirozenější formou zavádění IPv6 sítí je současný provoz obou protokolů. **Pokud je to jen trochu možné, je vhodné upřednostnit tuto variantu.** V tomto případě je nad IPv6 dobrá kontrola a v případě zavedení IPv6 touto formou se jedná už o finální řešení.

V místech, kde zatím není možné zavést IPv6 nativně, je možné využít některého z tunelovacích protokolů. V tomto případě je lepší věnovat čas konfiguraci buď ISATAP směrovače, anebo využít některého řešení prostřednictvím *Tunnel Brokeru* (například SixXS). Nejhorší variantu představují tunelovací mechanizmy vznikající automaticky (6to4, Teredo), které přináší velkou řadu komplikací v podobě obcházení bezpečnostních politik a nespolehlivého připojení. V současné době právě automatické tunelovací mechanizmy jsou zodpovědné za poměrně nízkou spolehlivost IPv6 připojení a obecně páchají víc škody než užitku. Potvrzuje to zpráva GEOFFA HUSTONA, Failing IPv6 , případně podobný výzkum laboratoří RIPE . Naštěstí se tyto mechanizmy uplatňují pouze při přístupu na služby, které jsou dostupné pouze po IPv6.

Zajímavým řešením se může zdát nasazení přechodových mechanizmů překladů protokolů v podobě NAT64, DNS64. Toto řešení předpokládá, že síť je budována pouze s využitím protokolu IPv6. Pokud ovšem nebudou dříve dořešeny záležitosti autokonfigurace (viz. IV díl seriálu) a připojování domácích a firemních sítí (viz. II. a III. díl seriálu), je toto řešení prakticky nepoužitelné. Všechny aplikace rovněž musí plnohodnotně podporovat IPv6, což zatím, zejména ve firemním sektoru, neplatí. Může se tedy jednat o řešení použitelné spíše v budoucnosti a dnes o něm nemá smysl příliš uvažovat.

Velice sporným problémem je automatické upřednostňování protokolu IPv6 před protokolem IPv4. Samotný fakt, že IPv6 dnes nefunguje vždy zcela spolehlivě, vede k tomu, že přístup na takové služby je z pohledu uživatele výrazně horší. V konečném důsledku vede uživatele spíše k tomu, že podporu IPv6 budou mít snahu spíše deaktivovat, a celkově vytváří velice negativní obraz o IPv6.

Tomáš Podermański

Autor pracuje jako správce metropolitní sítě VUT v Brně. Podílí se na řešení projektů zaměřenýchna bezpečnost a monitoring sítí. Je aktivním členem evropského projektu GÉAN3 v aktivitě Campus Best Practice.

Matěj Grégr

Studuje na Fakultě informačních technologií VUT v Brně. Snaží se porozumět počítačovým sítím a teoretické znalosti pak (ne)úspěšně uplatňovat v praxi jako správce kolejní sítě VUT.

Seriál Pohněme s IPv6

- IPv6 Mýty a skutečnost, díl V. - Zjednodušené hlavičky
- IPv6 Mýty a skutečnost, díl VI. - Bezpečnostní mechanizmy

- IPv6 Mýty a skutečnost, díl VII. - Podpora Multicast a anycast provozu
- IPv6 Mýty a skutečnost, díl VIII. - Přechodové mechanizmy
- IPv6 Mýty a skutečnost, díl IX. - Quo Vadis, IPv6?

Všechny díly seriálu