K A P I T O L A 13
Internet Protocol verze 6 (I Pv6)

V této kapitole se dozvíte základní informace o protokolu Internet Protocol verze 6 (lPv6). Z předchozích kapitol byste již měli mít slušný přehled o protokolu IPv4, ale pokud byste si chtěli dát opakování, nalistuje znovu kapitolu 3, "Tvorba podsítí, masky podsítí s proměnnou délkou (VLSM) a řešení problémů v TCP/IP". A kdybyste si náhodou nebyli jisti u výkladu adresních problémů týkajících se protokolu IPv4, měli byste si osvěžit kapitolu l l , "Překlady adres NAT (Network Address Translation)".
Protokol IPv6 se označuje jako "internetový protokol nové generace". Původně vznikl v reakci na nevyhnutelnou a hrozící krivylepšen, aby zajistil pružnost, efektivitu, možnosti a optimalizované funkce pro stále rostoucí požadavky uživatelů Internetu. Jeho předchůdce IPv4 se mu zdaleka nevyrovná, a proto nakonec skončí jako historická záležitost. Struktura hlavičky a adresy protokolu IPv6 jsou kompletně přepracovány a mnohé funkce, které byly u protokolu IPv4 v zásadě jen doplňkem a přílepkem, nyní u protokolu IPv6 tvoří součást základního standardu. Tento protokol je dobře vybaven a připraven k tomu, aby mohl zvládat těžko představitelné nároky budoucího Internetu. Pokusíme se tuto kapitolu pojmout co nejpřístupněji. Možná vás dokonce bude bavit, tak jako bylo potěšením ji psát. Protokol IPv6 je komplexní, ale zároveň elegantní, inovativní a plný funkcí. Mohli bychom jej přirovnat k nějakému futuristickému sportovnímu automobilu. Snad si tedy tuto kapitolu vychutnáte jako strhující jízdu! zi s vyčerpáním adres protokolu IPv4. O protokolu IPv6 jste pravděpodobně již slyšeli. Od počátečního návrhu byl dále

K čemu vlastně potřebujeme protokol Ipv6?

Stručná odpověď zní: protože potřebujeme komunikovat, a náš současný systém už nestačí - podobně jako koňská pošta nedokáže soutěžit s leteckou. Zamyslete se pouze nad tím, kolik času a úsilí musíme investovat do nových vynalézavých způsobů, jak šetřit šířku pásma a IP adresy. Ve své snaze o překonání stále horšího nedostatku adres jsme se dokonce dostali k maskám podsítí s proměnnou délkou (VLSM). Je to tak - každým dnem roste počet lidí a zařízení s připojením ke globální síti. To vůbec není špatné - neustále hledáme nové a vzrušující způsoby, jak komunikovat s dalšími osobami. V zásadě se jedná o přirozenou lidskou potřebu. Výhled do budoucna však není růžový. Jak jsme totiž zmínili v úvodu této kapitoly, použitelné adresy protokolu IPv4, na kterém v současnosti závisí naše schopnost komunikace v síti, brzy dojdou. Protokol IPv4 teoreticky dokáže poskytnout jen asi 4,3 miliardy adres, ale část z nich v praxi není dostupná. Ve skutečnosti lze zařízením přiřadit jen asi 250 milionů adres. Technologie beztřídního směrování mezi doménami (CIDR) a překladu adres NAT (Network Address Translation) sice pomáhají nevyhnutelný nedostatek adres oddálit, ale nakonec adresy přece jen dojdou a tento okamžik nastane během několika let. Čína se teprve začíná připojovat k Internetu a všichni víme, že spousta lidí a firem se tam už nemůže dočkat, až budou online. Různé studie docházejí k různým číslům. To, že se nejedná o žádné plané strašení, si však sami můžete ověřit prostou úvahou: na světě nyní žije asi 6,5 miliardy lidí a odhaduje se, že jen přes 10 procent populace je připojeno k Internetu!
Musíme se podívat pravdě do očí a přiznat, že vzhledem ke kapacitě protokolu IPv4 nemůže mít každý člověk svůj počítač - ani nemluvě o všech ostatních zařízeních, která do sítě připojujeme. Lidé, kteří pracují v oboru IT, zpravidla vlastní více počítačů. A to ani neuvažujeme telefony, notebooky, herní konzoly, faxy, směrovače, přepínače a hromadu dalších zařízení, která každodenně používáme. Nyní je tedy snad dostatečně zřejmé, že musíme něco udělat, než budou adresy vyčerpány a ztratíme schopnost se vzájemně připojovat, jak jsme byli zvyklí. Řešení představuje právě implementace protokolu IPv6.

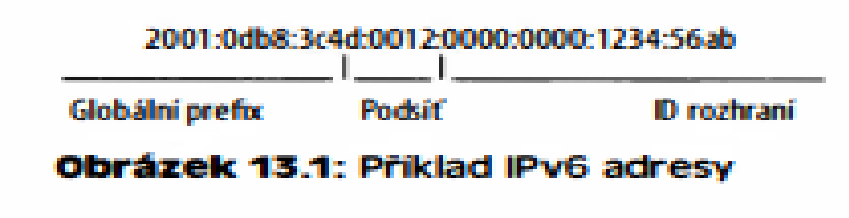
Výhody IPv6 a jeho využití

Je protokol IPv6 opravdu tak úžasný? Představuje opravdu odpověď na naše hrozící dilema? Skutečně stojí za upgrade z protokolu IPv4? To jsou vesměs dobré otázky a určitě by vás napadlo několik dalších. Někteří lidé se samozřejmě vyznačují starým známým "syndromem odporu ke změnám", ale nesmíte je poslouchat. Pokud by měli před lety hlavní slovo, stále bychom čekali týdny či měsíce, než naši poštu doručí listonoši na koních. Odpovědí na uvedené otázky je tedy jednoznačné ANO! Kromě toho, že protokol IPv6 poskytuje spoustu adres (3,4 x 1 01\38 = rozhodně dost), poskytuje v této verzi navíc mnoho dalších funkcí, díky kterým se náklady, čas i úsilí věnované na migraci každopádně vyplatí. Požadavky rozebereme v sekci "Přechod na IPv6" dále v této kapitole. Popíšeme si některé typy přechodu, které jsou potřebné při změně z verze 4 na verzi 6. Zároveň si ukážeme, že mimořádné výhody migrace značně převažují nad souvisejícími negativy.
Dnešní lokální sítě stejně jako celý Internet kladou mnoho nečekaných požadavků, které tvůrci protokolu IPv4 jednoduše nemohli předpokládat. Správci sítí se je pokoušejí kompenzovat pomocí mnoha doplňkových řešení, kvůli kterým je praktická implementace obtížnější, než kdyby je požadovaly síťové standardy. Protokol IPv6 mnoho těchto funkcí vylepšuje a zahrnuje je jako standardní a povinné. Mezi nové užitečné standardy patří protokol IPSec. Tato funkce poskytuje zabezpečení přenosu mezi koncovými zařízeními a budeme se jí zabývat v kapitole 1 4, "Rozlehlé sítě WAN". Další kouzelnou vlastností je mobilita. Jak je patrné z názvu, umožňuje přechod zařízení z jedné sítě do jiné bez výpadku konektivity.
Naprosto převratné jsou však funkce, které zvyšují efektivitu. V prvé řadě hlavička paketu IPv6 má polovinu polí a jsou zarovnána na hranice 64 bitů, což značně zvyšuje rychlost zpracování. V porovnání s protokolem IPv4 probíhá vyhledávání bleskovou rychlostí! Většina informací, které bývaly fixovány do hlavičky paketu IPv4, je nyní standardně vyjmuta. Podle svého uvážení je můžete do hlavičky vrátit formou volitelných rozšiřujících polí, která následují za základními poli hlavičky.
A pochopitelně zde máme celý nový vesmír plný adres (3,4 x 1 01\38), jak jsme se již zmínili. Kde se ale vzaly? Nalinkoval je snad nějaký Superman? Obrovský nárůst počtu adres přece musí mít nějakou příčinu! Protokol IPv6 značně zvětšuje adresní prostor prostě díky svému návrhu. Adresy jsou tedy mnohem delší - dokonce čtyřikrát. Adresa protokolu IPv6 má tedy délku 128 bitů. Žádné obavy, rozdělíme si ji na části a přesně si její strukturu vysvětlíme v nadcházející sekci "Adresy v IPv6 a jejich vyjadřování". Prozatím stačí zmínit, že prodloužení umožňuje vytvořit v adresním prostoru více hierarchických úrovní a zlepšit efektivitu architektury adres. Zvyšuje také efektivitu a škálovatelnost směrování, protože lze adresy mnohem účinněji agregovat. Protokol IPv6 také dovoluje, aby měli hostitelé a sítě více adres. To je důležité hlavně pro podniky, které usilují o co nejvyšší dostupnost. Nová verze protokolu IP nyní navíc zahrnuje rozšířené využití vícesměrové komunikace (kdy jedno zařízení vysílá mnoha hostitelům nebo vybrané skupině), což opět pomůže zlepšit efektivitu sítí, protože komunikace probíhá cíleněji.
Protokol IPv4 velmi často využívá všesměrové vysílání, což vede k mnoha problémům. Nejhorší z nich je samozřejmě obávaná všesměrová bouře - nekontrolovaná záplava předávaného všesměrového vysílání, která dokáže položit na kolena celou síť a pohltit poslední bity šířky pásma. Na všesměrovém provozu je navíc nepříjemné, že přerušuje činnost každého jednotlivého zařízení v síti. Při odeslání všesměrového vysílání musí každý počítač přestat s tím, co právě dělá, a odpovědět na zprávu, ať už je určena pro něj nebo nikoli. Dobrá zpráva tedy zní, že protokol IPv6 nic

takového jako všesměrové vysílání nezná, protože místo toho využívá provoz vícesměrového vysílání. K dispozici jsou navíc dva další typy komunikace: jednosměrové vysílání (unicast), které je obdobou stejné funkce u protokolu IPv4, a nový typ zvaný výběrová adresa (anycast). Komunikace na výběrovou adresu umožňuje přidělit stejnou adresu více zařízením. Když je tedy provoz odeslán na jedno zařízení adresované tímto způsobem, je předán nejbližšímu hostiteli, který sdílí stejnou adresu. To vše je pouze začátek - více informací o různých typech komunikace naleznete v sekci s názvem "Typy adres".

Adresy v IPv6 a jejich vyjadřování

Stejně jako při adresování v sítích IPv4 je znalost struktury a použití IP adres důležitá i v sítích IPv6. Již jsme uvedli, že adresy IPv6 jsou se svými 1 28 bity mnohem delší než adresy protokolu IPv4. Vzhledem k tomu a mnoha novým způsobům využití těchto adres lze usoudit, že bude protokol IPv6 nejspíš klást větší nároky na správu. Žádné obavy! Nyní se seznámíme se základními principy a ukážeme si, jak adresy vypadají, jak je lze zapisovat a jak se obvykle využívají. Výklad bude ze začátku možná trochu odtažitý, ale zakrátko vám vše přejde do krve. Podívejte se tedy na obrázek 1 3.1, kde je znázorněno rozdělení ukázkové IPv6 adresy do sekcí.



Obrázek 13.1: Příklad IPv6 adresy

Poznámka Pamatujte si polohu tohoto ID podsítě, protože ji budeme potřebovat v sekci "Konfigurace IPv6 v datové síti·· dále v této kapitole při konfiguraci směrovačů.

Nyní je tedy zřejmé, že adresa je mnohem větší. Co se kromě toho ještě změnilo? V prvé řadě si všimněte, že adresa obsahuje osm skupin číslic místo čtyř. Tyto skupiny jsou navíc místo tečkami odděleny dvojtečkami. A moment - adresa zahrnuje i písmena! Adresa je skutečně vyjádřena hexadecimálně stejně jako MAC adresa. Můžeme tedy říci, že adresa se skládá z osmi 1 6bitových hexadecimálních bloků oddělených dvojtečkami. To je úplný jazykolam, a to jsme ani nezkusili nahlas přečíst samotnou adresu!

Pokud si s protokolem IPv6 začnete hrát v testovací síti, musíte vědět o další vlastnosti jeho adres. Když pomocí webového prohlížeče navážete připojení protokolu HTTP k zařízení IPv6, musíte adresu na adresním řádku prohlížeče uzavřít do hranatých závorek. Proč? Dvojtečka se totiž v prohlížeči již používá k uvedení čísla portu. Kdybyste tedy adresu nezapsali v hranatých závorkách, prohlížeč by tuto informaci nedokázal interpretovat. Uveďme si příklad, jak to vypadá:

http : / /[2 001 : 0db8 : 3c4d : 0012 :0000:0000:1234:56a b] /default.html

Pokud to jen trochu půjde, určitě byste raději cílové umístění určili pomocí názvu (jako např. www.lammle.com). I když to však bude docela otravné, musíme se smířit s tím, že někdy bude potřeba zadat číselnou adresu. Je asi docela jasné, že služba DNS bude při implementaci protokolu IPv6 mimořádně důležitá.

Zkrácený zápis

Dobrá zpráva je, že při psaní těchto monstrózních adres si můžeme pomoci několika šikovnými triky. V praxi lze například vynechat části adresy a tím ji zkrátit. Můžeme si to však dovolit jen tehdy, budeme-Ii přitom dodržovat několik pravidel. V prvé řadě je možné vypustit úvodní nuly v každém jednotlivém bloku. Poté bude naše ukázková adresa vypadat takto:

2001 : db8 : 3c4d:1 2:0:0: 1234 : 56ab

To je rozhodně pokrok - když nic jiného, nemusíme psát všechny ty zbytečné nuly. Co však s celými bloky, které kromě nul nic jiného neobsahují? Můžeme svým způsobem vynechat i tyto bloky - alespoň některé. Když se vrátíme k naší ukázkové adrese, můžete odebrat dva bloky nul tak, že je nahradíme dvojicí dvojteček takto:

nee

Fajn - místo bloků plných nul jsme nyní dostali dvojici dvojteček. Aby nám to prošlo, nesmíme však v adrese nahradit více než jeden souvislý blok nul. Jestliže by tedy adresa obsahovala čtyři oddělené bloky nul, nemohli bychom je vypustit všechny. Pamatujte na pravidlo, že nahradit dvojicí dvojteček je možné jen jeden nepřerušený blok nul. Podívejte se na tento příklad:

2001 : 0 000 : 0000 : 0 0 12:0 000 : 0000 : 1 2 34 : 56ab

Uvědomte si, že nelze použít tento zápis:

2001 :: 12 :: 1 2 34 : 56ab

Místo toho můžete adresu maximálně zkrátit takto:

2001 :: 12:0:0: 1234 : 56ab

Výše uvedený zápis je nejlepší možný proto, že kdybychom odebrali dvě sady nul, zařízení analyzující adresu by nemělo dostatek informací, aby mohlo nuly vrátit. Směrovač by v praxi načetl nesprávnou adresu a poté by nemohl rozhodnout, zda má umístit dva bloky místo první dvojice dvojteček a dva místo druhé dvojice, nebo tři bloky místo prvního dvojitého symbolu a jeden blok místo druhého. Toto dilema by opravdu nebylo možné vyřešit, protože by adresa neobsahovala dostatek informací.

Typy adres

Již dobře známe jednosměrové, všesměrové a vícesměrové adresy protokolu IPv4, které v zásadě definují okruh komunikujících zařízení nebo alespoň jejich počet. Jak jsme však již zmínili, protokol IPv6 tuto trojicí rozšiřuje a přidává výběrovou adresu. Všesměrová vysílání, jak jsme na ně byli zvyklí, byla z protokolu IPv6 kvůli své neefektivnosti vyloučena. Podívejme se tedy, jak lze jednotlivé typy adresování a komunikace pomocí protokolu IPv6 využít.

Jednosměrová vysílání (unicast) - pakety adresované na jednosměrové adresy jsou doručeny jedinému rozhraní. U vyrovnávání zátěže může být stejnou adresou označeno více rozhraní. Existuje několik různých typů jednosměrových adres, ale zatím se nebudeme pouštět do podrobností.

Globální jednosměrové adresy - jedná se o běžně veřejně směrovatelné adresy stejného typu jako u protokolu IPv4.

Linkové lokální adresy - tyto adresy odpovídají privátním adresám protokolu IPv4 v tom, že se nepočítá s jejich směrováním. Můžeme je označit za šikovný nástroj, který dovoluje sestavit dočasnou síť LAN pro poradu nebo vytvořit malou lokální síť, která nebude podporovat směrování, ale přesto dovolí místní sdílení a přístup k souborům a službám.
Unikátní lokální adresy - tyto adresy jsou také určeny k jiným účelům než směrování, ale jsou téměř globálně jedinečné, takže málokdy nastane situace, kdy by se překrývaly s jinou sítí. Unikátní lokální adresy byly navrženy k tomu, aby nahradily síťové lokální adresy. Mají tedy přesně stejnou funkci jako privátní adresy u protokolu IPv4 - umožňují komunikaci v rámci lokality, ale zároveň dovolují směrování do více lokálních sítí. Síťové lokální adresy byly v září 2004 odmítnuty.
Vícesměrová vysílání (multicast) - opět podobně jako u protokolu IPv4 jsou pakety odeslané na adresu vícesměrového vysílání doručeny všem rozhraním, která vícesměrová adresa identifikuje. Tyto adresy se někdy označují jako adresy typu 1 :N. Vícesměrové adresy u protokolu IPv6 lze poznat velmi snadno, protože vždy začínají symboly FF. Podrobněji se o fungování vícesměrového vysílání zmíníme v sekci "Jak lPv6 funguje v datové síti".
Výběrová adresa (anycast) - stejně jako vícesměrové adresy také výběrové adresy označují více rozhraní, ale je zde zásadní rozdíl: paket výběrové adresy je doručen pouze na jedinou adresu - v praxi na první adresu, kterou s ohledem na směrovací vzdálenost nalezne. Tato adresa je přitom speciální, protože lze jedinou adresu aplikovat na více než jedno rozhraní. Mohli bychom je označit jako adresy typu l:l:N, ale termín "výběrová adresa" je mnohem srozumitelnější.
Pravděpodobně přemýšlíte nad tím, zda protokol IPv6 definuje nějaké speciální rezervované adresy, jaké existují u protokolu IPv4. Takové adresy zde jsou a není jich zrovna málo! Můžeme si je nyní projít.

Speciální adresy

Nyní si uvedeme některé adresy a adresní rozsahy, které byste si rozhodně měli pamatovat, protože je dříve nebo později budete potřebovat. Všechny jsou speciální nebo rezervované pro konkrétní použití, ale na rozdíl od protokolu IPv4 poskytuje protokol IPv6 celý vesmír adres. Když je jich proto pár vyhrazeno, vůbec nikomu nebudou chybět.
0:0:0:0:0:0:0:0 - odpovídá zápisu ::. Jedná se o ekvivalent IPv4 adresy 0.0.0.0 a obvykle se jedná o zdrojovou adresu hostitele při použití stavové konfigurace.
0:0:0:0:0:0:0:1 - odpovídá zápisu :: 1. Ekvivalent adresy 1 27.0.0. 1 v protokolu IPv4.
0:0:0:0:0:0:192.168.100.1 - takto bychom zapsali IPv4 adresu ve smíšeném síťovém prostředí protokolů IPv6/IPv4.
2000::/3 - adresní rozsah globálního jednosměrového vysílání.
FCOO::/7 - rozsah unikátního lokálního jednosměrového vysílání.
FE80::/1O - rozsah linkového lokálního jednosměrového vysílání.
FFOO::/8 - rozsah vícesměrového vysílání.
3FFF:FFFF::/32 - vyhrazeno pro příklady a dokumentaci.
2001:0DB8::/32 - rovněž vyhrazeno pro příklady a dokumentaci.
2002::/16 - používá se u 6to4, což je přechodový systém - struktura, která umožňuje přenos paketů IPv6 po síti IPv4 bez nutnosti výslovně konfigurovat tunely.
K tomuto tématu se znovu vrátíme v sekci "Přechod na IPv6", ale prozatím si ukážeme, jak protokol IPv6 v datové síti prakticky funguje. Všichni víme, jak funguje protokol IPv4, takže se zaměříme na novinky.

Jak IPv6 funguje v datové síti

Je načase prozkoumat jednotlivé aspekty protokolu IPv6. Přitom je vhodné začít ukázkou, jak adresovat hostitele a jakým způsobem může tento hostitel najít jiného hostitele a prostředky v síti. Ukážeme si také, že zařízení mohou automaticky adresovat sama sebe, což se označuje jako bezstavová automatická konfigurace. Spolu s tím existuje i další typ konfigurace - tzv. stavová konfigurace. Nezapomínejte, že stavová automatická konfigurace využívá server DHCP velmi podobným způsobem jako při konfiguraci protokolu IPv4. Také si předvedeme, jak v síti IPv6 funguje protokol ICMP (Internet Control Message Protocol) a vícesměrové vysílání.

Automatická konfigurace

Automatická konfigurace představuje mimořádně užitečné řešení, protože umožňuje zařízením v síti, aby si samostatně přidělila linkovou lokální jednosměrovou adresu. Tento proces začíná prvním načtením informace o prefixu ze směrovače a pokračuje přidáním vlastní adresy rozhraní zařízení jako ID rozhraní. Kde však zařízení dostane ID rozhraní? Jak víte, každé zařízení v síti Ethernet má fyzickou MAC adresu a to je právě ten údaj, který slouží jako ID rozhraní. ID rozhraní v IPv6 adrese má však délku 64 bitů a MAC adresa je dlouhá pouze 48 bitů. Odkud tedy pochází zbývajících 16 bitů? MAC adresa je doplněna dodatečnými bity uprostřed. Používá se přitom sekvence FFFE.
Řekněme například, že máme zařízení s MAC adresou v tomto tvaru: 0060.d673.1987. Po doplnění bude vypadat následovně: 0260.d6FF.FE73. 1 987. Odkud se tedy vzalo číslo 2 na začátku adresy? Další dobrá otázka. V rámci procesu prodloužení adresy (na tzv. upravený formát eui-64) se mění bit, který určuje, zda je adresa lokálně nebo globálně unikátní. Mění se přitom sedmý bit v adrese. Bitová hodnota 1 znamená globálně jedinečnou a bitová hodnota O označuje lokálně jedinečnou adresu. Je tedy adresa v uvedené ukázce globálně či lokálně jedinečná? Správná odpověď zní, že je globálně jedinečná. Tato funkce šetří čas při adresování hostitelských počítačů, které tento úkol zajišťují samostatně na základě komunikace se směrovačem. Při automatické konfiguraci hostitel využívá základní postup se dvěma kroky:
l. Hostitel při konfiguraci svého rozhraní potřebuje nejdříve informaci o prefixu (podobná síťové části IPv4 adresy). Odešle tedy směrovači příslušný požadavek RS (router solicitation). Tento požadavek RS je následně odeslán jako vícesměrové vysílání na všechny vícesměrové adresy směrovače. V praxi se tato zpráva odesílá pomocí protokolu ICMP a jako všechna síťová data má i tato zpráva ICMP číslo, které ji identifikuje. Zpráva RS má typ ICMP 1 33.
2. Směrovač odpoví požadovanou informací o prefixu ve formě zprávy RA (router advertisement). Zpráva RA má také formát vícesměrového paketu, který směřuje na vícesměrovou adresu každého uzlu a má typ ICMP 1 34. Zprávy RA se odesílají pravidelně, ale hostitel vyžaduje na zprávu RS okamžitou odpověď. Nemusí tedy kvůli potřebným informacím čekat na další plánovanou zprávu RA.

Tyto dva kroky jsou znázorněny na obrázku 13.2.

Obrázek 13.2: Dva kroky při automatické konfiguraci protokolu IPv6

Tento typ automatické konfigurace se mimochodem označuje jako bezstavová automatická konfigurace, protože se při tom jiné zařízení nekontaktuje, nepřipojuje se k němu a nepřijímá od něj žádné další informace. Ke stavové konfiguraci se zakrátko dostaneme v diskusi o protokolu DHCPv6. Podívejme se nyní, jak nakonfigurovat protokol IPv6 u směrovačů Cisco.

Konfigurace IPv6 na směrovačích Cisco

Chcete-Ii u směrovače povolit protokol IPv6, musíte zadat příkaz globální konfigurace ipv6 unicast-routing:
Corp(config)#ipv6 unicast- routing
Ve výchozím nastavení je předávání provozu IPv6 zakázáno, takže tímto příkazem je povolíte. Jak jste také pravděpodobně již uhodli, protokol IPv6 není standardně zapnut ani na žádném rozhraní. Je tedy potřeba přejít na každé rozhraní zvlášť a postupně jej povolit.
Lze to provést několika způsoby, ale skutečně snadný postup spočívá v pouhém přidání adresy na rozhraní. Zajistíte to příkazem i pv6 address < ipv6prefix>/ [eui -64].
Uveďme si příklad:
Corp (config-ifl #i pv6 address 2001 : db8 : 3c4d:I:0260.d 6FF. FE73 . 1987/64
Můžete uvést celou 1 28bitovou globální IPv6 adresu, nebo můžete použít možnost eui-64. Vzpomeňte si, že formát eui-64 dovoluje, aby zařízení použilo svou vlastní MAC adresu a jejím doplněním získalo ID rozhraní. Vyzkoušejme si to:
Corp(config-if)#i pv6 address 2 0 0 1 : d b8 : 3c4d:1::/ 64 eui-64
Místo zadávání IPv6 adresy u směrovače můžete povolit rozhraní, aby bylo možné použít automatickou linkovou lokální adresu.
Poznámka Pamatujte, že máte-Ii pouze linkovou lokální adresu, dokážete komunikovat pouze v dané lokální podsíti.
Chcete-li nakonfigurovat směrovač tak, aby používal pouze linkové lokální adresy, uveďte příkaz konfigurace rozhraní i pv6 enable:
Corp (config-if)# ipv6 enable
Pusťme se nyní do stavové konfigurace protokolu IPv6 a nakonfigurujme server DHCP tak, aby byl kompatibilní s protokolem IPv6.

Protokol DHCPv6

Protokol DHCPv6 funguje velmi podobně jako DHCP u protokolu IPv4 - samozřejmě s tím rozdílem, že podporuje nové adresní schéma IPv6. Možná vás překvapí, ale protokol DHCP nadále poskytuje mnoho dalších možností, které nejsou u automatické konfigurace k dispozici. Skutečně - v případě automatické konfigurace není ani slechu po serverech DNS, doménových názvech či mnoha dalších funkcích, které protokol DHCP prostřednictvím protokolu IPv4 vždy zajišťoval. To je hlavní důvod, proč budete protokol DHCP většinou s protokolem IPv6 používat i nadále.
Při spuštění v režimu IPv4 odešle klient zprávu DHCP Discover, aby vyhledal server, který mu poskytne požadované informace. Pamatujte však, že u protokolu IPv6 nejdříve probíhají procesy RS a RA. Pokud je do sítě připojen server DHCPv6, oznámí zpráva RA vrácená klientovi, zda je DHCP možné použít. Pokud není nalezen směrovač, klient odpoví odesláním zprávy DHCP Solicit. Jedná se v zásadě o vícesměrovou zprávu s požadavkem se zdrojovou adresou ff 02:: 1 :2, což označuje všechny agenty DHCP - jak běžné, tak synchronizační servery.
Systém Cisco lOS naštěstí zajišťuje určitou podporu protokolu DHCPv6. Je však omezena na bezstavový server DHCP, tzn. nenabízí žádnou správu adres ve fondu. Možnosti konfigurace tohoto fondu adres jsou navíc omezeny pouze na servery DNS, doménové servery a servery SIP. To znamená, že určitě budete potřebovat další server, který dokáže poskytovat a distribuovat všechny další požadované informace a zaj išťovat správu přiřazených adres. Přesto si uveďme konfiguraci bezstavového serveru DHCP v systému lOS směrovače - konfigurace se velmi podobá obdobné konfiguraci u protokolu IPv4:

```
Router1(config)#ipv6 dhcp pool ?
WORD DHCP pool name
Router1(config)#ipv6 dhcp pool test
Router1(config-dhcp)#?
IPv6 DHCP configuration commands:
  default                    Set a command to its defaults
  dns-server                 DNS servers
  domain-name                Domain name to complete unqualified host names
  exit                       Exit from DHCPv6 configuration mode
  no                         Negate a command or set its defaults
  prefix-delegation          IPv6 prefix delegation
  sip                        SIP Servers options
Router1(config-dhcp)#dns-server ?
  Hostname or X:X:X:X::X     Server's name or IPv6 address
Router1(config-dhcp)#domain-name lammle.com
Router1(config-dhcp)#prefix-delegation ?
  X:X:X:X::X/<0-128>         IPv6 x:x::y/<z>
  aaa                        Acquire prefix from AAA
  pool                       IPv6 prefix pool
Router1(config-dhcp)#prefix-delegation pool ?
  WORD IPv6 prefix pool
Router1(config-dhcp)#prefix-delegation pool test ?
  lifetime Configure prefix lifetimes
  <cr>
Router1(config-dhcp)#prefix-delegation pool test lifetime ?
  <60-4294967295>            Valid lifetime (seconds)
  at                         Expire prefix at a specific time/date
  infinite                   Infinite valid lifetime
Router1(config-dhcp)#prefix-delegation pool test lifetime 3600 ?
  <60-4294967295>            Preferred lifetime (seconds)
  infinite                   Infinite preferred lifetime
Router1(config-dhcp)#prefix-delegation pool test lifetime 3600 3600 ?
  <cr>
Router1(config-dhcp)#prefix-delegation pool test lifetime 3600 3600
```

Všimněte si, že jako u protokolu DHCP v kombinaci s IPv4 není nutné nastavit časový limit. Když jsme není nakonfigurovali fond, stačí jej přiřadit rozhraní, což je odchylka od protokolu IPv4:

```
Router1(config)#int fa 0/0
Router1(config-if)#ipv6 dhcp server ?
  WORD Name of IPv6 DHCP pool
Router1(config-if)#ipv6 dhcp server test
```

Nyní máme kompletně nakonfigurovaný server DHCPv6, který je přiřazen rozhraní fa0/0.

Protokol ICMPv6

Protokol IPv4 využívá ICMP k mnoha účelům, jak např. přenos chybových zpráv typu nedosažitelnosti cíle a funkcím pro řešení potíží typu příkazů Ping a Traceroute. Protokol ICMPv6 tyto služby poskytuje i nadále, ale na rozdíl od svého předchůdce není verze 6 implementována jako samostatný protokol vrstvy 4. Jedná se o integrovanou součást protokolu IPv6 a přenáší se za základní hlavičkou protokolu IPv6 jako rozšiřující hlavička. Protokol ICMPv6 navíc přidává další skvělou funkci - zabraňuje jakékoli fragmentaci protokolu IPv6 díky procesu ICMPv6, který se označuje jako zjišťování trasy MTU.

Funguje to následovně: Zdrojový uzel připojení odešle paket, který odpovídá velikosti MTU pro MTU lokální linky. Když je tento paket přenášen po trase k cíli, vynutí každá linka s menší hodnotou MTU než velikost aktuálního paketu, aby mezilehlý směrovač odeslal zpět zdrojovému počítači zprávu "paket je příliš velký". Zpráva sděluje zdrojovému uzlu maximální velikost povolenou restriktivní linkou a požádá zdroj, aby odeslal nový zmenšený paket, který lze po příslušné lince přenést. Tento proces pokračuje až do konečného dosažení cíle. V této fázi již zdrojový uzel používá hodnotu MTU nové trasy. Zbytek přenášených datových paketů je tedy poté chráněn před fragmentací.

Protokol ICMPv6 nyní přebírá úlohu zjišťování adresy dalších zařízení na lokální lince. U protokolu IPv4 tuto funkci zastával protokol ARP (Address Resolution Protocol), ale u ICMPv6 se nyní nazývá Neighbor Discovery. Tento proces je založen na vícesměrové adrese označované jako adresa požadujícího uzlu. Všichni hostitelé se při připojení k síti připojí do této všesměrové skupiny. Část jejich IPv6 adresy (24 bitů zcela vpravo) se doplní na konec vícesměrové adresy FF02:0:0:0:0: 1 :FF/1 04. Při dotazu na tuto adresu odešle odpovídající hostitel zpět svou adresu vrstvy 2. Zařízení mohou vyhledávat a sledovat další sousední zařízení v síti velmi podobným způsobem. V diskusi o zprávách RA a RS v předchozí části kapitoly jsme uvedli, že k požadavkům a odesílání informací adresy používají vícesměrový provoz. Jedná se rovněž o funkci protokolu ICMPv6 - konkrétně zjišťování sousedů.

U protokolu IPv4 se používal protokol IGMP, který umožňoval hostitelskému zařízení zjistit svůj místní směrovač, jenž se připojoval do vícesměrové skupiny a požadoval příjem provozu pro danou skupinu. Tato funkce protokolu IGMP byla nahrazena protokolem ICMPv6 a proces byl přejmenován na zjišťování vícesměrového naslouchání (multicast Iistener discovery).

Směrovací protokoly nad IPv6

Většina směrovacích protokolů, kterými jsme se již zabývali, byla upgradována pro použití v sítích IPv6. Mnohé funkce a konfigurace, které jsme dosud uvedli, se také uplatňují téměř stejným způsobem. Vzhledem k tomu, že byla z protokolu IPv6 odstraněna všesměrová vysílání, je zřejmé, že protokoly odkázané výhradně na všesměrový provoz skončí na smetišti dějin. Těchto monster, která pohlcují šířku pásma a ničí výkon sítě, nebude vůbec škoda. Směrovací

protokoly nadále používané s protokolem IPv6 mají nový název a vylepšený vzhled. Podíváme se na ně v následujících odstavcích.
Jako první je na řadě protokol RIPng (next generation). Pokud se v oboru IT již nějakou dobu pohybujete, víte, že protokol RIP fungoval velmi dobře v malých sítích. Proto unikl likvidaci a zůstává k dispozici i v éře protokolu IPv6. Kromě toho je dostupný i protokol EIGRPv6, protože již obsahuje moduly závislé na protokolu a stačí přidat nový modul pro protokol lPv6. Skupinu přeživších protokolů uzavírá OSPFv3. Není to překlep, opravdu se jedná o verzi 3. OSPF pro IPv4 byl ve skutečnosti druhou verzí, takže se po upgradu na IPv6 změnil v protokol OSPFv3.

Protokol RIPng

Primární funkce protokolu RIPng jsou v praxi stejné jako ve verzi RIPv2. I nadále se jedná o protokol s vektorem vzdáleností, má maximální počet přeskoků 15 a používá rozdělení horizontu, znehodnocenou zpětnou aktualizaci a další mechanismy předcházení smyčkám. Nyní však používá port 52 1 protokolu UDP. Pořád také odesílá své aktualizace pomocí vícesměrového vysílání, ale u protokolu IPv6 používá transportní adresu FF02::9. To je docela elegantní, protože ve verzi RIPv2 se jednalo o vícesměrovou adresu 224.0.0.9. Adresa v novém vícesměrovém rozsahu protokolu IPv6 tedy opět končí devítkou. Většina směrovacích protokolů si v praxi obdobně uchovává část své identity z doby protokolu IPv4.
Pokud by se však nová verze nevyznačovala žádnými rozdíly, těžko by se dala označit za novou verzi. Víme, že směrovače ve svých směrovacích tabulkách uchovávají adresu dalšího přeskoku svých sousedních směrovačů pro každou cílovou síť. Rozdíl spočívá v tom, že u protokolu RIPng sleduje směrovač tuto adresu dalšího přeskoku pomocí linkové lokální adresy, nikoli globální adresy. Pravděpodobně jedna z největších změn u protokolu RIPng (a v podstatě u všech směrovacích protokolů IPv6) spočívá v tom, že se zveřejňování sítě konfiguruje či povoluje z režimu konfigurace rozhraní, nikoli síťovým příkazem v režimu konfigurace směrovače. Jestliže tedy protokol RIPng povolíte přímo na rozhraní a nepřejdete do režimu konfigurace směrovače se spuštěním procesu RIPng, bude nový proces RIPng jednoduše spuštěn automaticky. Vypadá to asi takto:
Routerl1 (confi g-if)#ipv6 rip 1 enable
Číslo 1 uvedené v příkazu je značka, která identifikuje spuštěný proces RIPng. Jak jsme již uvedli, zajistí spuštění procesu RIPng, takže není nutné přecházet do režimu konfigurace směrovače. I nadále je však možné přejít do režimu konfigurace směrovače z jiného důvodu, např. kvůli konfiguraci jiné funkce typu redistribuce. V tomto případě bude příkaz směrovače vypadat takto:
Router1 (config)#ipv6 router rip 1
Router1 (config-rtr)#
Pamatujte tedy, že protokol RIPng funguje velmi podobně jako u protokolu IPv4. Hlavní rozdíl leží v tom, že nyní používá vlastní síť místo síťového příkazu, kterým se dříve povolovalo rozhraní pro směrování do připojené sítě.

Protokol EIGRPv6

Tak jako protokol RIPng i protokol EIGRPv6 funguje v zásadě stejně jako jeho předchůdce určený pro IPv4 a k dispozici zůstává většina funkcí, které protokol EIGRP poskytoval před příchodem verze EIGRPv6. EIGRPv6 je stále pokročilým protokolem s vektorem vzdáleností, jenž má některé funkce protokolu se stavem linky. Proces zjišťování sousedů pomocí zpráv Hello probíhá i nadále a protokol stále poskytuje spolehlivou komunikaci založenou na stabilním transportním protokolu, který zajišťuje rychlou konvergenci bez smyček na základě difuzního aktualizačního algoritmu DUAL.
Pakety Hello a aktualizace se přenášejí pomocí vícesměrového vysílání a stejně jako u protokolu RIPng zůstává obdobná vícesměrová adresa protokolu EIGRPv6. U protokolu IPv4 se jednalo o adresu 224.0.0. 10 a ve verzi IPv6 se používá adresa FF02::A (A = lO v hexadecimální notaci). Mezi oběma verzemi však pochopitelně existují rozdíly. Hlavní změna spočívá v tom, že stejně jako u protokolu RIPng odpadá použití síťového příkazu. Síť i rozhraní pro zveřejnění je nutné povolit z režimu konfigurace rozhraní. Stále je však potřeba povolit směrovací protokol EIGRPv6 v režimu konfigurace směrovače, protože proces směrování je nutné doslovně zapnout stejně jako rozhraní příkazem no shutdown.
Konfigurace protokolu EIGRPv6 bude vypadat takto:

Konfigurace protokolu EIGRPv6 bude vypadat takto:

```
Router1(config)#ipv6 router eigrp 10
```

Číslo 10 v tomto případě opět představuje číslo autonomního systému (AS). Výzva se změní do tvaru `(config-rtr)` a poté je potřeba zadat příkaz `no shutdown`:

```
Router1(config-rtr)#no shutdown
```

V tomto režimu lze také konfigurovat jiné možnosti typu redistribuce.

Přejděme tedy k rozhraní a povolme protokol IPv6:

```
Router1(config-if)#ipv6 eigrp 10
```

Hodnota 10 v příkazu rozhraní znovu odkazuje na číslo autonomního systému, který byl povolen v režimu konfigurace.

Posledním protokolem z této skupiny je směrovací protokol OSPF kompatibilní s protokolem IPv6.

Protokol OSPFv3

Nová verze protokolu OSPF neporušuje pravidlo, že se směrovací protokoly vyznačují mnoha podobnými vlastnostmi jako jejich verze pro protokol IPv4. Základy protokolu OSPF zůstávají beze změny - i nadále se jedná o směrovací protokol se stavem linky, který dělí celou datovou síť nebo autonomní systém na oblasti za vzniku hierarchie. Můžete si gratulovat, že OSPF s více oblastmi je alespoň prozatím mimo rámec okruhů zkoušky CCNA. Několik možností, které jsme rozebírali v kapitole 7, "Protokoly EIGRP (Enhanced IGRP) a OSPF (Open Shortest Path First)", se však poněkud liší.
Ve verzi 2 protokolu OSPF je ID směrovače (RID) určeno nejvyšší IP adresou přiřazenou směrovači (případně je lze přiřadit ručně). Ve verzi 3 se nastavuje hodnota RID, ID oblasti a ID stavu linky, což jsou vesměs statické 32bitové hodnoty. Již je však není možné zjistit pomocí IP adresy, protože IPv6 adresa má 128 bitů. Změny týkající se přiřazení

těchto hodnot spolu s odebráním informací IP adresy z hlaviček paketů protokolu OSPF způsobují, že novou verzi protokolu OSPF lze směrovat pomocí téměř libovolného protokolu síťové vrstvy. Atributy přilehlosti a dalšího přeskoku nyní využívají linkové lokální adresy a protokol OSPFv3 nadále odesílá své aktualizace a potvrzení pomocí vícesměrového vysílání. Adresa FF02::5 je přitom určena pro směrovače OSPF a adresa FF02::6 pro vyhrazené směrovače OSPF. Tyto nové adresy nahrazují dřívější adresy 224.0.0.5 a 224.0.0.6.
Jiné a méně pružné protokoly IPv4 neposkytují možnosti, které protokol OSPFv2 nabízí při přiřazení konkrétních sítí a rozhraní v procesu OSPF. Tato konfigurace však nadále probíhá v rámci procesu konfigurace směrovače. U protokolu OSPFv3 se stejně jako u ostatních směrovacích protokolů IPv6, o kterých jsme se zmínili, konfigurují rozhraní, a tedy i sítě k nim připojené přímo na rozhraní v režimu jeho konfigurace. Konfigurace protokolu OSPFv3 bude vypadat takto:

Konfigurace protokolu OSPFv3 bude vypadat takto:

```
Router1(config)#ipv6 router osfp 10
Router1(config-rtr)#router-id 1.1.1.1
```

Některá nastavení typu sumarizace a redistribuce je potřeba provést v režimu konfigurace směrovače, ale při konfiguraci protokolu OSPFv3 z rozhraní již není potřeba jej konfigurovat z příkazového řádku.

Po dokončení konfigurace rozhraní je proces konfigurace směrovače přidán automaticky a konfigurace rozhraní vypadá takto:

```
Router1(config-if)#ipv6 ospf 10 area 0.0.0.0
```

Stačí tedy přejít ke každému rozhraní a přiřadit ID procesu a oblast.

Se znalostí uvedených informací můžeme přejít k výkladu o migraci na protokol IPv6 z verze IPv4.

<u>Přechod na IPv6</u>
Již jsme se obšírně zabývali fungováním protokolu IPv6 a jeho konfigurací v síti. Jaké však budou náklady jeho zavedení? A kolik práce s tím bude spojeno? To jsou jistě dobré otázky, ale odpověď bude pro každého jiná. Náklady totiž budou značně záviset na tom, jaké prvky již infrastruktura obsahuje. Je jasné, že pokud jste se snažili prodloužit morální životnost starých směrovačů a přepínačů a nyní musíte kvůli kompatibilitě s protokolem IPv6 všechny upgradovat, může být změna docela rozsáhlá. A to zatím neuvažujeme o operačních systémech na serverech a v pracovních stanicích a o práci, úsilí a dokonce námaze při zajištění kompatibility všech aplikací. Náklady tedy mohou být poměrně vysoké! Dobrá zpráva je, že pokud jste skutečně nepřestali sledovat vývoj, je již několik roků většina operačních systémů a síťových zařízení s protokolem IPv6 kompatibilní. Jeho funkce se pouze zatím nevyužívaly. Zbývá však zodpovědět dotaz týkající se pracnosti a času. Na rovinu je potřeba uvést, že to může být docela dřina. V každém případě bude nějakou dobu trvat, než se vám všechny systémy podaří převést a než zkontrolujete, že vše správně funguje. Jestliže se jedná o velkou síť se spoustou zařízení, může přechod trvat opravdu dlouho. Nelekejte se však - aby bylo možné provést integraci pomaleji, vznikly příslušné migrační strategie. Nyní si ukážeme tři primární strategie přechodu, které jsou k dispozici. První se označuje jako duální sady protokolů a umožňuje, aby zařízení používalo současně sadu protokolů IPv4 i IPv6. Může tedy pokračovat ve stávající komunikaci a zároveň pracovat s novou komunikací protokolu IPv6 tak, jak byl tento protokol implementován. Další strategie se nazývá tunelování typu 6t04. Tato volba je nejvhodnější, máte-li síť, kde se používá výhradně protokol IPv6, a potřebujete prostřednictvím sítě IPv4 komunikovat s jinou sítí IPv6. Třetí možnost si necháme jako překvapení na konec.
Duální sady protokolů
Tento typ migrační strategie se využívá nejčastěji, protože je prostě nejsnazší - umožňuje zařízením komunikovat pomocí protokolu IPv4 nebo IPv6. Duální sady protokolů dovolují upgradovat zařízení a aplikace v síti postupně. S rostoucím počtem upgradovaných hostitelů a zařízení v síti se zvyšuje podíl komunikace pomocí protokolu IPv6. Nakonec vše funguje nad protokolem IPv6 a lze odebrat staré sady protokolu IPv4, které již nejsou potřeba. Konfigurace duálních sad protokolů u směrovače Cisco je navíc překvapivě snadné - stačí zapnout předávání protokolu IPv6 a aplikovat adresy na rozhraní, která jsou již nastavena pro protokol IPv4. Vypadá to asi takto:

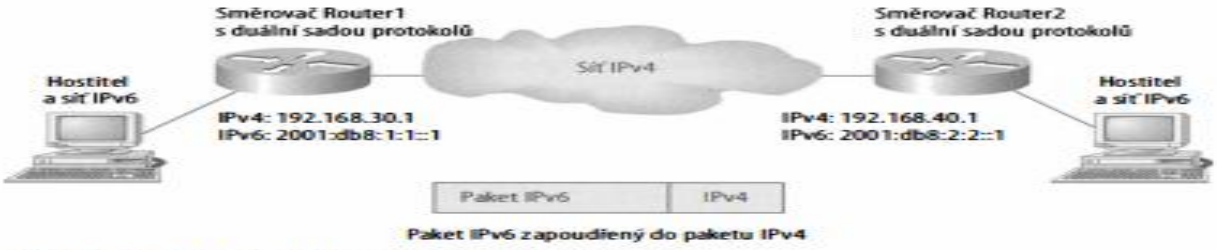
```
Corp(config)#ipv6 unicast-routing
Corp(config)#interface fastethernet 0/0
Corp(config-if)#ipv6 address 2001:db8:3c4d:1::/64 eui-64
Corp(config-if)#ip address 192.168.255.1 255.255.255.0
```

Po pravdě řečeno je však docela dobré rozumět různým postupům tunelování, protože doba, kdy bude protokol IPv6 jediným směrovaným protokolem, asi hned tak nenastane.

<u>Tunelování typu 6to4</u>
Tunelování typu 6t04 je mimořádně užitečné při přenosu dat IPv6 po síti, která stále používá protokol IPv4. Je docela možné, že máte podsítě protokolu IPv6 nebo jiné části sítě, které pracují výhradně s protokolem IPv6, a tyto oblasti musí vzájemně komunikovat. To není tak složité, ale když uvážíte, že k tomu může docházet po spojích WAN nebo jiné síti, nad kterou nemáte kontrolu, mohou se objevit určité potíže. Jak to tedy vyřešit, když nemáme kontrolu nad celou trasou? Můžeme vytvořit tunel, který bude provoz protokolu IPv6 přenášet přes síť IPv4. Celá koncepce tunelování není příliš složitá a tunely lze v praxi vytvářet docela snadno. Stačí přitom jen zachytit paket IPv6, který putuje po síti, a připojit k němu hlavičku protokolu IPv4. Trochu to připomíná sportovní rybaření, s tou výjimkou, že ryba puštěná zpět do vody obvykle nemá nic přilepeno na hlavě.
Lépe si vše můžete představit při pohledu na obrázek 1 3.3. Aby však vše fungovalo, vyžaduje to několik směrovačů s duální sadou protokolů, což jsme si právě ukázali. Nyní je potřeba doplnit krátké konfigurační příkazy, aby mezi těmito směrovači vznikl tunel. Tunely jsou docela jednoduché - stačí jen sdělit každému směrovači, kde tunel začíná a kde by měl končit. Na základě obrázku 1 3.3 nakonfigurujeme tunel u každého směrovače:

```
Router1(config)#int tunnel 0
Router1(config-if)#ipv6 address 2001:db8:1:1::1/64
Router1(config-if)#tunnel source 192.168.30.1
Router1(config-if)#tunnel destination 192.168.40.1
Router1(config-if)#tunnel mode ipv6ip

Router2(config)#int tunnel 0
Router2(config-if)#ipv6 address 2001:db8:2:2::1/64
Router2(config-if)#tunnel source 192.168.40.1
Router2(config-if)#tunnel destination 192.168.30.1
Router2(config-if)#tunnel mode ipv6ip
```



Obrázek 13.3: Vytvoření tunelu typu 6to4

V tomto schématu mohou sítě IPv6 nyní komunikovat přes síť IPv4. Na tomto místě je potřeba uvést, že tato konfigurace není myšlena jako trvalá. Konečným cílem by měl být provoz celkové a úplné sítě IPv6 mezi koncovými body. Musíme poznamenat, že pokud by síť IPv4, která v této situaci slouží pro přenos, obsahovala bod překladu NAT, právě vytvořené tunelové zapouzdření by selhalo! Překlad adres NAT byl v průběhu let do značné míry upgradován, takže nyní dokáže obsluhovat konkrétní protokoly a dynamické připojení. Bez jednoho z těchto upgradů překlad adres NAT obvykle většinu spojení přeruší. Vzhledem k tomu, že tato migrační strategie není součástí většiny implementací překladu adres NAT, hrozí při tom problémy.
Lze je však obejít pomocí řešení Teredo, které dovoluje přenést veškerý tunelový provoz do paketů protokolu UDP. Překlad adres NAT s pakety UDP nemanipuluje, takže nejsou porušeny jako pakety jiných protokolů. Pokud tedy funguje řešení Teredo a pakety jsou ukryty v plášti protokolu UDP, mohou překladem adres NAT projít nepoškozeny.

Překlady protokolů NAT-PT

Pravděpodobně jste již slyšeli, že protokol IPv6 nezahrnuje žádný překlad adres NAT. Tato informace je správná - do jisté míry. Sám o sobě protokol IPv6 implementaci překladu adres NAT neobsahuje. To je však pravda pouze formálně, protože existuje strategie přechodu označovaná jako překlady protokolů NAT (NAT protocol translation - NAT-PT). Uvědomte si pouze, že tento přístup je vhodné použít pouze tehdy, není-li jiná možnost, protože není zrovna optimální. Hostitelé IPv4 při tom mohou komunikovat pouze s jinými hostiteli IPv4 a hostitelé v nativní síti IPv6 s jinými hostiteli IPv6. Co to znamená? U přístupu založeném na tunelování jsme vzali pakety IPv6 a zamaskovali jsme je jako pakety IPv4. V případě překladů protokolů NAT-PT žádné zapouzdření neexistuje - data zdrojového paketu jsou vyjmuta z paketu jednoho typu a znovu zabalena jako paket IP nového cílového typu. Konfigurace překladů protokolů NAT-PT je sice mimo rámec okruhů zkoušky CCNA, ale přesto si ji vysvětlíme. Stejně jako u překladu adres NAT pro IPv4 1ze implementaci provést několika způsoby.
Statické překlady protokolů NAT-PT poskytují mapování typu 1:1 z jediné IPv4 adresy na jedinou IPv6 adresu (připomíná to statický NAT). Existují také dynamické překlady protokolů NAT-PT, které pomocí fondu IPv4 adres zajišťují mapování typu 1:1 s IPv6 adresou (opět je to velmi povědomé). Nakonec jsou k dispozici překlady protokolů NAPT-PT (Network Address Port Translation), kdy funguje mapování více IPv6 adres na jednu IPv4 adresu a číslo portu (je zřejmé, v čem spočívá rozdíl oproti překladu adres NAT). Jak je patrné, překlad adres NAT neslouží k převodu veřejných a privátních IPv6 adres, jako tomu bylo u protokolu IPv4, ale místo toho mezi protokoly IPv4 a IPv6. Opět je nutné připomenout, že tato metoda by se měla používat pouze jako poslední možnost. Ve většině případů funguje přístup založený na tunelování mnohem lépe a bez komplikace s konfigurací a systémové režie.

Konfigurace IPv6 v datové síti

V této sekci nakonfigurujeme datovou síť, kterou jsme používali v celé knize (síť obsahuje pět propojených směrovačů). Nebudeme však přidávat protokol IPv6 u směrovače 87 1W ani do sítí LAN a WLAN připojených ke směrovačům Rl, R2 a R3, aby byla konfigurace jednodušší a srozumitelnější. Nejdříve tedy přidáme protokol IPv6 ke směrovačům Corp, Rl, R2 a R3. Poté doplníme směrovací protokoly RIP a OSPF a na závěr si předvedeme několik příkazů na ověření konfigurace. Jako obvykle začneme směrovačem Corp:

Jako obvykle začneme směrovačem Corp:

```
Corp#config t
Corp(config)#ipv6 unicast-routing
Corp(config)#int f0/1
Corp(config-if)#ipv6 address 2001:db8:3c4d:11::/64 eui-64
Corp(config-if)#int s0/0/0
Corp(config-if)#ipv6 address 2001:db8:3c4d:12::/64 eui-64
Corp(config-if)#int s0/0/1
Corp(config-if)#ipv6 address 2001:db8:3c4d:13::/64 eui-64
Corp(config-if)#int s0/1/0
Corp(config-if)#ipv6 address 2001:db8:3c4d:14::/64 eui-64
Corp(config-if)#int s0/2/0
Corp(config-if)#ipv6 address 2001:db8:3c4d:15::/64 eui-64
Corp(config-if)#^Z
Corp#copy run start
Destination filename [startup-config]?[enter]
Building configuration...
```

```
[OK]
Corp#
```

V předchozí konfiguraci jsme pouze mírně upravili adresu podsítě každého rozhraní. Podívejme se na směrovací tabulku:

```
Corp#sh ipv6 route
IPv6 Routing Table - 12 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route I1 - ISIS L1, I2 - ISIS L2, IA - ISIS
       interarea, IS - ISIS summary O - OSPF intra, OI - OSPF inter,
       OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
C  2001:DB8:3C4D:11::/64 [0/0]
     via ::, FastEthernet0/1
L  2001:DB8:3C4D:11:21A:2FFF:FE55:C9E9/128 [0/0]
     via ::, FastEthernet0/1
C  2001:DB8:3C4D:12::/64 [0/0]
     via ::, Serial0/0/0
L  2001:DB8:3C4D:12:21A:2FFF:FE55:C9E8/128 [0/0]
     via ::, Serial0/0/0
C  2001:DB8:3C4D:13::/64 [0/0]
     via ::, Serial0/0/1
L  2001:DB8:3C4D:13:21A:2FFF:FE55:C9E8/128 [0/0]
     via ::, Serial0/0/1
C  2001:DB8:3C4D:14::/64 [0/0]
     via ::, Serial0/1/0
L  2001:DB8:3C4D:14:21A:2FFF:FE55:C9E8/128 [0/0]
     via ::, Serial0/1/0
C  2001:DB8:3C4D:15::/64 [0/0]
     via ::, Serial0/2/0
L  2001:DB8:3C4D:15:21A:2FFF:FE55:C9E8/128 [0/0]
     via ::, Serial0/2/0
L  FE80::/10 [0/0]
     via ::, Null0
L  FF00::/8 [0/0]
     via ::, Null0
Corp#
```

Co tedy znamenají dvě adresy pro každé rozhraní? U jedné je uvedeno C (connected) a jedna je označena písmenem L. Připojená adresa je IPv6 adresa nakonfigurovaná na každém rozhraní a adresa s písmenem L je automaticky přiřazená linková lokální adresa. U linkové lokální adresy si všimněte, že je do ní vložena sekvence FF:FE, aby vznikla adresa typu eui-64. Než přejdeme ke směrovači RI, musíme zmínit ještě jednu věc. Všimněte si, že při adresování rozhraní jsme k číslu podsítě každého z nich přidali odlišné číslo. Nepřehlédněte také, že tato čísla kopírují privátní IPv4 adresy. To zjednodušuje správu. Pusťme se tedy do konfigurace směrovače RI:

```
R1#config t
R1(config)#ipv6 unicast-routing
R1(config)#int s0/0/0
R1(config-if)#ipv6 address 2001:db8:3c4d:12::/64 eui-64
```

```
R1(config-if)#int s0/0/1
R1(config-if)#ipv6 address 2001:db8:3c4d:13::/64 eui-64
R1(config-if)#^Z
R1#show ipv6 route
IPv6 Routing Table - 6 entries
[kódy jsou zkráceny]
C 2001:DB8:3C4D:12::/64 [0/0]
  via ::, Serial0/0/0
L 2001:DB8:3C4D:12:21A:6DFF:FE64:9B2/128 [0/0]
  via ::, Serial0/0/0
C 2001:DB8:3C4D:13::/64 [0/0]
  via ::, Serial0/0/1
L 2001:DB8:3C4D:13:21A:6DFF:FE64:9B2/128 [0/0]
  via ::, Serial0/0/1
L FE80::/10 [0/0]
  via ::, Null0
L FF00::/8 [0/0]
  via ::, Null0
R1#
```

Všimněte si, že na každé straně linky se používají přesně stejné IPv6 adresy podsítě. Nakonfigurujme směrovače R2 a R3 a poté přidejme protokol RIPv6:

```
R2#config t
R2(config)#ipv6 unicast-routing
R2(config)#int s0/2/0
R2(config-if)#ipv6 address 2001:db8:3c4d:14::/64 eui-64
R2(config-if)#do show ipv6 route
IPv6 Routing Table - 4 entries

C 2001:DB8:3C4D:14::/64 [0/0]
  via ::, Serial0/2/0
L 2001:DB8:3C4D:14:213:60FF:FE20:4E4C/128 [0/0]
  via ::, Serial0/2/0
L FE80::/10 [0/0]
  via ::, Null0
L FF00::/8 [0/0]
  via ::, Null0
R2(config-if)#
To vypadá dobře. Přejděme ke směrovači R3:
R3#config t
R3(config)#ipv6 unicast-routing
R3(config)#int s0/0/1
R3(config-if)#ipv6 address 2001:db8:3c4d:15::/64 eui-64
R3(config-if)#do show ipv6 route
IPv6 Routing Table - 4 entries

C 2001:DB8:3C4D:15::/64 [0/0]
  via ::, Serial0/0/1
L 2001:DB8:3C4D:15:21A:6DFF:FE37:A44E/128 [0/0]
  via ::, Serial0/0/1
L FE80::/10 [0/0]
```

```
    via ::, Null0
L FF00::/8 [0/0]
    via ::, Null0
R3(config-if)#
```

Znovu si všimněte, že na každé straně linky od směrovače Corp ke směrovačům R1, R2 a R3 se používají přesně stejné IPv6 adresy podsítě. Nyní můžeme začít přidávat směrovací protokoly!

Konfigurace protokolu RIPng

Tato část je skutečně snadná – stačí přejít ke každému rozhraní u všech směrovačů a zadat jeden příkaz. Pusťme se do toho:

```
Corp#config t
Corp(config)#int f0/1
Corp(config-if)#ipv6 rip ?
WORD User selected string identifying this RIP process
Corp(config-if)#ipv6 rip 1 enable
Corp(config-if)#int s0/0/0
Corp(config-if)#ipv6 rip 1 enable
Corp(config-if)#int s0/0/1
Corp(config-if)#ipv6 rip 1 enable
Corp(config-if)#int s0/1/0
Corp(config-if)#ipv6 rip 1 enable
Corp(config-if)#int s0/2/0
Corp(config-if)#ipv6 rip 1 enable
```

Nastavme směrovač R1:

```
R1#config t
R1(config)#int s0/0/0
R1(config-if)#ipv6 rip 1 enable
R1(config-if)#int s0/0/1
R1(config-if)#ipv6 rip 1 enable
Konfigurace směrovače R2:
R2#config t
R2(config)#int s0/2/0
R2(config-if)#ipv6 rip 1 enable
Konfigurace směrovače R3:
R3#config t
R3(config)#int s0/0/1
R3(config-if)#ipv6 rip 1 enable
```

Je na čase ověřit směrovací tabulky a konfiguraci protokolu IPv6.

Kontrola konfigurace RIPng

Začneme běžným příkazem `show ip route`. Následuje výstup směrovače R3:

```
R3#sh ipv6 route
R 2001:DB8:3C4D:11::/64 [120/2]
    via FE80::21A:2FFF:FE55:C9E8, Serial0/0/1
R 2001:DB8:3C4D:12::/64 [120/2]
```

```
    via FE80::21A:2FFF:FE55:C9E8, Serial0/0/1
R   2001:DB8:3C4D:13::/64 [120/2]
    via FE80::21A:2FFF:FE55:C9E8, Serial0/0/1
R   2001:DB8:3C4D:14::/64 [120/2]
    via FE80::21A:2FFF:FE55:C9E8, Serial0/0/1
C   2001:DB8:3C4D:15::/64 [0/0]
    via ::, Serial0/0/1
L   2001:DB8:3C4D:15:21A:6DFF:FE37:A44E/128 [0/0]
    via ::, Serial0/0/1
L   FE80::/10 [0/0]
    via ::, Null0
L   FF00::/8 [0/0]
    via ::, Null0
R3#
```

To připomíná běžnou tabulku RIP protokolu IPv4, včetně administrativní vzdálenosti a počtu přeskoků. Uvedeny jsou podsítě 11, 12, 13, 14 a 15.

Podívejme se na několik dalších testovacích příkazů:

```
R3#sh ipv6 protocols
IPv6 Routing Protocol is "connected"
IPv6 Routing Protocol is "static"
IPv6 Routing Protocol is "rip 1"
  Interfaces:
    Serial0/0/1
  Redistribution:
    None
R3#
```

Příkaz `show ipv6 protocols` moc informací neposkytuje. Zkusme příkaz `show ipv6 rip`:

```
R3#sh ipv6 rip
RIP process "1", port 521, multicast-group FF02::9, pid 60
  Administrative distance is 120. Maximum paths is 16
  Updates every 30 seconds, expire after 180
  Holddown lasts 0 seconds, garbage collect after 120
  Split horizon is on; poison reverse is off
  Default routes are not generated
  Periodic updates 44, trigger updates 19
Interfaces:
  Serial0/0/1
Redistribution:
  None
```

To vypadá sdílněji. Vidíme, že administrativní vzdálenost se nadále rovná 120. Kromě toho je uvedena vícesměrová skupina, maximální trasy a časovače. Pokračujme tedy dvěma dalšími kontrolními příkazy. Prvním z nich je příkaz `show ipv6 interface s0/0/1`:

```
R3#sh ipv6 interface serial 0/0/1
Serial0/0/1 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::21A:6DFF:FE37:A44E
  Global unicast address(es):
    2001:DB8:3C4D:1:21A:6DFF:FE37:A44E, subnet is 2001:DB8:3C4D:1::/64 [EUI]
```

```
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::9
    FF02::1:FF37:A44E
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds
  Hosts use stateless autoconfig for addresses.
```

V tomto výpise lze najít další docela užitečná data. Nejlepší nás však teprve čeká: příkaz `debug ipv6 rip` vypadá slibně:

```
R3#debug ipv6 rip
*May 24 18:31:11.959: RIPng: Sending multicast update on Serial0/0/1 for 1
*May 24 18:31:11.959: src=FE80::21A:6DFF:FE37:A44E
*May 24 18:31:11.959: dst=FF02::9 (Serial0/0/1)
*May 24 18:31:11.959: sport=521, dport=521, length=32
*May 24 18:31:11.959: command=2, version=1, mbz=0, #rte=1
*May 24 18:31:11.959: tag=0, metric=1, prefix=2001:DB8:3C4D:1::/64
*May 24 18:40:44.079: %LINEPROTO-5-UPDOWN: Line protocol on Interface
  Serial0/0/0, changed state to down
*May 24 18:31:24.959: RIPng: response received from
FE80::21A:2FFF:FE55:C9E8 on Serial0/0/1 for 1
*May 24 18:31:24.959: src=FE80::21A:2FFF:FE55:C9E8 (Serial0/0/1)
*May 24 18:31:24.959: dst=FF02::9
*May 24 18:31:24.959: sport=521, dport=521, length=32 *May 24 18:31:24.959:
  command=2, version=1, mbz=0, #rte=1
*May 24 18:31:24.959: tag=0, metric=16, prefix=2001:DB8:3C4D:12::/64
*May 24 18:31:24.959: RIPng: 2001:DB8:3C4D:12::/64, path
FE80::21A:2FFF:FE55:C9E8/Serial0/0/1 unreachable
*May 24 18:31:24.959: RIPng: 2001:DB8:3C4D:12::/64, expired, ttg is 120
*May 24 18:31:24.959: RIPng: Triggered update requested
*May 24 18:31:25.959: RIPng: generating triggered update for 1
*May 24 18:31:25.959: RIPng: Suppressed null multicast update on
Serial0/0/1 for 1
```

To je opravdu zajímavé. Je zřejmé, že pracujeme se zdrojovým i cílovým portem 52 1 (stále používáme protokol UDP) a síť či podsíť 12 není dosažitelná. Rozhraní sO/O/O směrovače Corp se totiž právě rozhodlo začít stávkovat. (Psaní této knihy je docela tvrdá práce!) V každém případě je zřejmé, že protokol RIPng se i nadále vyznačuje některými základními vlastnostmi protokolu RIP pro IPv4. V dalším kroku nastavíme u směrovačů podporu protokolu OSPFv3.

Konfigurace protokolu OSPFv3

Stejně jako u konfigurace protokolu RIPng se protokol OSPF v datové síti povoluje na každém rozhraní, kde má fungovat. . Konfigurace směrovače Corp vypadá takto:

```
Corp#config t
Corp(config)#int f0/1
Corp(config-if)#ipv6 ospf 1 ?
  area Set the OSPF area ID
Corp(config-if)#ipv6 ospf 1 area 0
Corp(config-if)#int s0/0/1
Corp(config-if)#ipv6 ospf 1 area 0
Corp(config-if)#int s0/1/0
Corp(config-if)#ipv6 ospf 1 area 0
Corp(config-if)#int s0/2/0
Corp(config-if)#ipv6 ospf 1 area 0
Corp(config-if)#^Z
Corp#
```

To není zlé – dokonce trochu snazší než u protokolu IPv4. Nakonfigurujme zbývající tři směrovače:

```
R1#config t
R1(config)#int s0/0/1
R1(config-if)#ipv6 ospf 1 area 0
R1(config-if)#
*May 24 19:24:55.279: %OSPFv3-5-ADJCHG: Process 1, Nbr 172.16.10.2 on
  Serial0/0/1 from LOADING to FULL, Loading Done
```

Směrovač R1 je nyní sousedem směrovače Corp. Zajímavý je řádek výstupu, který sděluje, že se při změně přilehlosti protokolu OSPFv3 používá identifikátor RID protokolu IPv4.

```
R2#config t
R2(config)#int s0/2/0
R2(config-if)#ipv6 ospf 1 area 0
R2(config-if)#
*May 24 19:27:31.399: %OSPFv3-5-ADJCHG: Process 1, Nbr 172.16.10.3 on
  Serial0/1/0 from LOADING to FULL, Loading Done
```

Znovu se objevila informace o přilehlosti , což je skvělé. Ještě jeden směrovač a poté začneme s kontrolou:

```
R3#config t
R3(config)#int s0/0/1
R3(config-if)#ipv6 ospf 1 area 0
R3(config-if)#
*May 24 19:29:07.231: %OSPFv3-5-ADJCHG: Process 1, Nbr 172.16.10.4 on
  Serial0/2/0 from LOADING to FULL, Loading Done
```

Zatím jsme síť sice neprověřili, ale zdá se, že vše funguje správně. Přesto však kontrolu nesmíme vynechat!

Kontrola konfigurace OSPFv3

Začneme jako obvykle příkazem `show ipv6 route`:

```
R3#sh ipv6 route IPv6
Routing Table - 7 entries
O 2001:DB8:3C4D:11::/64 [110/65]
  via FE80::21A:2FFF:FE55:C9E8, Serial0/0/1
```

```
O 2001:DB8:3C4D:13::/64 [110/128]
   via FE80::21A:2FFF:FE55:C9E8, Serial0/0/1
O 2001:DB8:3C4D:14::/64 [110/128]
   via FE80::21A:2FFF:FE55:C9E8, Serial0/0/1
C 2001:DB8:3C4D:15::/64 [0/0]
   via ::, Serial0/0/1
L 2001:DB8:3C4D:15:21A:6DFF:FE37:A44E/128 [0/0]
   via ::, Serial0/0/1
L FE80::/10 [0/0]
   via ::, Null0
L FF00::/8 [0/0]
   via ::, Null0
R3#
```

Vidíme všechny podsítě (kromě podsítě 12, která je kvůli vadnému rozhraní vypnuta). Vyzkoušejme příkaz `show ipv6 protocols`**:**

```
R3#sh ipv6 protocols
IPv6 Routing Protocol is "connected"
IPv6 Routing Protocol is "static"
IPv6 Routing Protocol is "rip 1"
  Interfaces:
    Serial0/0/1
  Redistribution:
    None
IPv6 Routing Protocol is "ospf 1"
  Interfaces (Area 0):
    Serial0/0/1
  Redistribution:
None
```

V případě dalšího příkazu přejdeme zpět ke směrovači Corp, abychom mohli zobrazit více připojení: `show ipv6 ospf neighbor`.

```
Corp#sh ipv6 ospf neighbor
Neighbor ID     Pri       State     Dead Time     ID rozhrani    Interface
172.16.10.4     1         FULL/     - 00:00:36    3              Serial0/2/0
172.16.10.3     1         FULL/     - 00:00:33    16             Serial0/1/0
172.16.10.2     1         FULL/     - 00:00:30    6              Serial0/0/1
Corp#
```

Moment! Potřebujeme zadat příkazy pro ladění. Použijeme dva z nich: `debug ipv6 ospf packet` **a** `debug ipv6 ospf hello` **(téměř shodné s příkazy pro protokol IPv4):**

```
Corp#debug ipv6 ospf packet
  OSPFv3 packet debugging is on
Corp#
*May 24 19:38:12.283: OSPFv3: rcv. v:3 t:1 l:40 rid:172.16.10.3
     aid:0.0.0.0 chk:E1D2 inst:0 from Serial0/1/0
Corp#
*May 24 19:38:15.103: OSPFv3: rcv. v:3 t:1 l:40 rid:172.16.10.4
     aid:0.0.0.0 chk:7EBB inst:0 from Serial0/2/0
Corp#
```

```
*May 24 19:38:18.875: OSPFv3: rcv. v:3 t:1 l:40 rid:172.16.10.2
     aid:0.0.0.0 chk:192D inst:0 from Serial0/0/1
Corp#
*May 24 19:38:22.283: OSPFv3: rcv. v:3 t:1 l:40 rid:172.16.10.3
     aid:0.0.0.0 chk:E1D2 inst:0 from Serial0/1/0
Corp#un all
All possible debugging has been turned off
Corp#debug ipv6 ospf hello
OSPFv3 hello events debugging is on
Corp#
*May 24 19:38:32.283: OSPFv3: Rcv hello from 172.16.10.3 area 0 from
     Serial0/1/0 FE80::213:60FF:FE20:4E4C interface ID 16
*May 24 19:38:32.283: OSPFv3: End of hello processing
Corp#
*May 24 19:38:35.103: OSPFv3: Rcv hello from 172.16.10.4 area 0 from
     Serial0/2/0 FE80::21A:6DFF:FE37:A44E interface ID 6
*May 24 19:38:35.103: OSPFv3: End of hello processing
Corp#
*May 24 19:38:38.875: OSPFv3: Rcv hello from 172.16.10.2 area 0 from
     Serial0/0/1 FE80::21A:6DFF:FE64:9B2 interface ID 6
*May 24 19:38:38.875: OSPFv3: End of hello processing
Corp#un all
All possible debugging has been turned off
Corp#
```

Parádní výstup! Určitě se shodneme, že tato kapitola byla velmi zajímavá. Dokonce jsme se setkali s poruchou rozhraní jako v praxi. Problematika protokolu IPv6 je skutečně atraktivní. Chcete-li tento protokol zvládnout, sežente pokud možno nějaké směrovače a pusťte se do testování!

Shrnutí

V této kapitole jsme si vysvětlili základy protokolu IPv6 a zprovoznění tohoto protokolu v datovém síti Cisco. Jak již víte, i pouhá analýza a základní konfigurace vyžadují rozsáhlé znalosti - a v této kapitole jsme zůstali jen na povrchu. Přesto však víte více, než budete potřebovat ke zvládnutí okruhů zkoušky CCNA. Nejdříve jsme popsali důvody pro zavedení protokolu IPv6 a výhody, které jsou s ním spojeny. Poté jsme přešli k popisu adresování u protokolu IPv6 a použití zkrácených výrazů. V rámci diskuse o adresování u protokolu IPv6 jsme si předvedli různé typy adres spolu se speciálními adresami, které jsou u protokolu IPv6 vyhrazeny. Protokol IPv6 se obvykle zavádí automaticky, což znamená, že hostitelé využívají automatickou konfiguraci. Rozebrali jsme proto použití automatické konfigurace protokolem IPv6 a její uplatnění při nastavení směrovače Cisco. Poté jsme si ukázali, jak u směrovače přidat server DHCP, aby mohl poskytovat možnosti hostitelům - nikoli adresy, ale možnosti typu adresy serveru DNS.
Protokol ICMP má u protokolu IPv6 mimořádný význam. Podrobně jsme vysvětlili, jak protokol ICMP s protokolem IPv6 spolupracuje a poté jsme se zabývali konfigurací protokolů RIP, EIGRP a OSPF s protokolem IPv6. Ani přechod na protokol IPv6 není maličkost a ukázali jsme si jeho výhody a nevýhody spolu se třemi migračními strategiemi - duální sadou protokolů, tunelováním pomocí protokolů IPv4 i IPv6 a třetí přístup zvaný NAT-PT, který je vhodné použít jen jako poslední variantu. Nakonec jsme se dostali k postupům konfigurace protokolu IPv6 v datové síti, kterou používáme v rámci celé této knihy, a demonstrovali jsme, jak ověřit konfiguraci pomocí různých příkazů show, jež jsou u protokolu IPv6 k dispozici.

Klíčové poznatky ke zkoušce

Měli byste vědět, proč potřebujeme protokol IPv6 - bez protokolu IPv6 by celosvětově došly IP adresy.
Seznamte se s linkovými lokálními adresami - linková lokální adresa připomíná privátní adresu u protokolu IPv4, ale nelze ji směrovat, dokonce ani v rámci podnikové sítě.
Seznamte se s unikátními lokálními adresami - tyto adresy podobně jako linkové lokální adresy připomínají privátní adresy protokolu IPv4 a neumožňují směrování v Internetu. Rozdíl mezi linkovou lokální a unikátní lokální adresou však spočívá v tom, že unikátní lokální adresu je možné v rámci organizační či podnikové sítě směrovat.
Pamatujte si principy adresování protokolu IPv6 - adresování protokolu IPv6 se liší od protokolu IPv4. Adresování protokolu IPv6 využívá mnohem větší adresní prostor. Adresy mají délku 128 bitů a vyjadřují se hexadecimálně na rozdíl od adres protokolu IPv4, které jsou dlouhé jen 32 bitů a uvádějí se pomocí desítkových čísel.