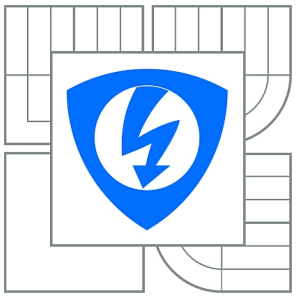
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

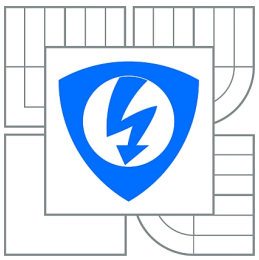
NÁVRH NOVÝCH LABORATORNÍCH ÚLOH PRO PROSTŘEDÍ GNS3

SEMESTRÁLNÍ PRÁCE
SEMESTRAL THESIS

AUTOR PRÁCE — Bc. MARTIN BARNIAK
AUTHOR

VEDOUCÍ PRÁCE — Ing. JAN JEŘÁBEK, Ph.D.
SUPERVISOR

BRNO 2014



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav telekomunikací

Semestrální práce

magisterský navazující studijní obor
Telekomunikační a informační technika

Student:	Bc. Martin Barniak	***ID:***	136498
Ročník:	2	***Akademický rok:***	2014/2015

NÁZEV TÉMATU:

Návrh nových laboratorních úloh pro prostředí GNS3

POKYNY PRO VYPRACOVÁNÍ:

Prostudujte možnosti simulačního prostředí GNS3/Dynamips, problematiku operačních systémů pro Cisco zařízení a obsahy kurzů XCA3 až XCA5. V rámci této diplomové práce se zaměřte na návrh čtyř nových laboratorních úloh určených pro virtualizované prostředí. Hlavní témata úloh volte zejména z těchto okruhů: porovnání protokolů IPv4 a IPv6 při různých operacích, směrovací protokoly s IPv4 a IPv6, problematika síťové bezpečnosti s IPv4 a IPv6, technika MPLS a tunelovací protokoly. Ke každé úloze vytvořte i český návod vhodný pro studenty, včetně kontrolních otázek a doplňujících úkolů. Kde to bude vhodné, připravte do úlohy výchozí topologii a konfiguraci. Délka každé úlohy musí být přibližně dvě a půl hodiny. V rámci semestrálního projektu detailněji rozpracujte dvě úlohy.

DOPORUČENÁ LITERATURA:

[1] TEARE, Diane. Implementing Cisco IP routing (ROUTE): foundation learning guide : foundation learning for the ROUTE 642-902 exam. Indianapolis: Cisco Press, 2010, xxix, 945 s. ISBN 978-1-58705-882-0.
[2] FROOM, Richard, Balaji SIVASUBRAMANIAN a Erum FRAHIM. Implementing Cisco IP switched networks (SWITCH): foundation learning guide. 1st ed. Indianapolis: Cisco Press, 2010, xxiv, 526 s. ISBN 978-1-58705-884-4.
[3] RANJBAR, Amir. Troubleshooting and mainting cisco IP networks (TSHOOT) foundation learning guide: foundation learning for the CCNP TSHOOT 642-832. 1st ed. Indianapolis: Cisco Press, 2010, xviii, 531 s. ISBN 978-1-58705-876-9.

Termín zadání:	23.9.2014	***Termín odevzdání:***	16.12.2014

Vedoucí práce: Ing. Jan Jeřábek, Ph.D.
Konzultanti semestrální práce:

doc. Ing. Jiří Mišurec, CSc.
Předseda oborové rady

ABSTRAKT

Diplomová práca sa zaoberá štyrmi laboratórnymi úlohami v simulačnom prostredí GNS3. Navrhnuté úlohy sa primárne zameriavajú porovnaním protokolov IPv4 a IPv6. V prvej úlohe sa vyskytuje tematika smerovacích protokolov OSPFv2 a OSPFv3. Ďalej úloha obsahuje tranzitné techniky ako NAT–PT a tunelovanie typu GRE a 6to4. Druhá úloha sa zaoberá konfiguráciou smerovacích protokolov EIGRP a EIGRPv6. Ďalej pojednáva o tematike protokolov DHCP a ICMP v rámci protokolovej sady IPv4 a IPv6. Tretia úloha sa primárne zaoberá bezpečnosťou protokolovej sady IPv6. Obsahuje autentizáciu OSPFv3, prístupové zoznamy a stavový Cisco IOS firewall. Obsahom štvrtej úlohy je protokol MPLS. Prvá časť úlohy je venovaná základnej konfigurácií protokolu MPLS a v druhej časti je úloha zameraná na MPLS v IPv6. Všetky úlohy v sebe zahŕňajú kontrolné otázky a samostatnú úlohu.

KĽÚČOVÉ SLOVÁ

GNS3, IPv4, IPv6, OSPFv2, OSPFv3, EIGRP, EIGRPv6 DHCP, ICMP, IPsec, IOS Firewall, MPLS, modelovanie sietí

ABSTRACT

Diploma thesis deals with four laboratory tasks in simulation environment GNS3. Designed tasks are primarily focused on comparison of IPv4 and IPv6 protocols. In the first task the subject is concerned about OSPFv2 and OSPFv3 routing protocols. Next themes are transit techniques like NAT–PT and tunneling like GRE and 6to4. The second task is focused on configuration of routing protocols like EIGRP and EIGRPv6. Next sections are concerned about DHCP and ICMP protocols within IPv4 and IPv6 protocol suits. The third task is primarily focused on security relations of protocol suite IPv6. It contains OSPFv3 authentication, access lists and Cisco stateful IOS firewall. Content of the fourth task is protocol MPLS. First part of this task is concerned about basic configuration of this protocol and second part is focused on MPLS within IPv6 environment. All tasks contain test questions and individual part task.

KEYWORDS

GNS3, IPv4, IPv6, OSPFv2, OSPFv3, EIGRP, EIGRPv6 DHCP, ICMP, IPsec, IOS Firewall, MPLS, network modeling

BARNIAK, Martin *Návrh nových laboratorních úloh pro prostředí GNS3*: diplomová práca. BRNO: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2015. 140 s. Vedúci práce bol Ing. Jan Jeřábek, Ph.D.

PREHLÁSENIE

Prehlasujem, že som svoju diplomovoú prácu na tému „Návrh nových laboratorních úloh pro prostředí GNS3“ vypracoval samostatne pod vedením vedúceho diplomovej práce, s využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej diplomovej práce ďalej prehlasujem, že v súvislosti s vytvorením tejto diplomovej práce som neporušil autorské práva tretích osôb, najmä som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a/alebo majetkových a som si plne vedomý následkov porušenia ustanovenia § 11 a nasledujúceho autorského zákona č. 121/2000 Sb., o autorských právach, o právach súvisiacich s právom autorským a o zmene niektorých zákonov (autorský zákon), v znení neskorších predpisov, vrátane možných trestnoprávnych dôsledkov vyplývajúcich z ustanovenia časti druhej, hlavy VI. diel 4 Trestného zákoníka č. 40/2009 Sb.

BRNO ....

(podpis autora)

POĎAKOVANIE

Touto cestou by som sa veľmi rád poďakoval môjmu vedúcemu práce pánovi Ing. Janovi Jeřábkovi, Ph.D. za odborné vedenie, konzultácie, trpezlivosť a podnetné návrhy k práci.

BRNO ....

(podpis autora)

SIX research centre
sensor, information and communication systems

Faculty of Electrical Engineering
and Communication
Brno University of Technology
Purkynova 118, CZ-61200 Brno
Czech Republic
http://www.six.feec.vutbr.cz

POĎAKOVANIE

Výzkum popísaný v této diplomovej práci bol realizovaný v laboratóriách podporených z projektu SIX; registračné číslo CZ.1.05/2.1.00/03.0072, operačný program Výzkum a vývoj pre inováce.

BRNO ....

(podpis autora)

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI

2007–13
OP Výzkum a vývoj pro inovace

OBSAH

ZOZNAM OBRÁZKOV

ZOZNAM TABULIEK

ÚVOD

Počítačové siete resp. sieťové technológie sú neoddeliteľnou súčasťou modernej doby. Ich využitie si nájde miesto viac či menej prakticky v každom odvetví. Sieťové technológie sa stali tak komplexnými, že vo väčšine prípadov nestačí sieť iba navrhnúť a následne implementovať. Toto by mohlo mať za následok neočakávané chovanie a dokonca pád celej už funkčnej siete. Mohla by nastať situácia, že nami zvolené neodsimulované riešenie by malo katastrofálne následky z ekonomického pohľadu. V ostrej produkcií niekoľko minútový výpadok komunikačnej siete môže spôsobiť niekoľko stá tisícové škody. Jednou z možností ako riešiť tento problém ponúkajú simulačné programy. Existuje niekoľko simulačných programov, pomocou ktorých sme schopní virtualizovať časť alebo aj celú počítačovú sieť. Toto riešenie je časovo efektívne a cenovo nenáročné v porovnaní s priamym nasadením prvkov do reálnej siete. V rámci simulačného programu môžeme merať rôzne parametre a analyzovať počítačovú sieť v vzhľadom na možnosti daného simulačného nástroja.

Cieľom tejto diplomovej práce je vytvorenie laboratórnych úloh v prostredí GNS3 (Graphical Network Simulator 3), ktoré sú primárne zamerané na porovnanie protokolov IPv4 (Internet Protocol version 4) a IPv6 (Internet Protocol version 6). Úlohy vychádzajú z teórie, ktorá je uvedená v teoretickej časti práce. Návody sú písané v českom jazyku kvôli lepšiemu porozumeniu pre českého študenta.

Účelom prvej kapitoly je stručné predstavenie simulačného prostredia GNS3. Kapitola popisuje emulátor Dynamips. Ďalej je v kapitole obsiahnutý popis funkcií pre efektívne hospodárenie s výpočetnými zdrojmi počas práce v prostredí GNS3.

Druhá kapitola obsahuje stručný popis problematiky IOS (Internetwork Operating System) operačných systémov pre smerovače a prepínače od firmy Cisco.

Tretia kapitola zhŕňa základné body, ktoré charakterizujú protokolovú sadu IPv4. Obsahuje stručný popis protokolu, adresáciu a štruktúru paketu. Na konci sa nachádza rozbor protokolov ICMP a DHCP pre túto protokolu sadu.

Štvrtá kapitola obsahuje charakteristiku protokolovej sady IPv6. V tejto kapitole je obsiahnutý základný popis protokolu, adresácia, štruktúra paketu, rozbor protokolov ICMPv6 (Internet Control Message Protocol version 6) a DHCPv6 (Dynamic Host Configuration Protocol version 6). Ďalej kapitola obsahuje techniky migrácie IPv4 a IPv6 a záver je venovaný bezpečnosti v IPv6, pričom kapitola zachytáva náležitosti týkajúce sa laboratórnej úlohy číslo 3.

Piata kapitola obsahuje stručný popis IGP (Interior Gateway Protocol) smerovacích protokolov, ktoré sa používajú v rámci autonómneho systému. Záver tejto kapitoly porovnáva rozdiely IGP smerovacích protokolov IPv4 a ich ekvivalentov v IPv6.

Šiesta kapitola stručne pojednáva o protokole MPLS (Multiprotocol Label Switching) a technikách, ktoré umožňujú využitie tohto protokolu v IPv6.

Siedma kapitola obsahuje vlastnú prácu. Obsahuje prehľadný popis a prácu v prostredí GNS3. Popisuje teoretický rozbor laboratórnych úloh. Samotné návody k úlohám sa nachádzajú v prílohe diplomovej práce. Úlohy sú koncipované v zmysle „krok za krokom“ no treba podotknúť, že sa od študenta očakáva istá vedomosť danej problematiky. Náročnosť laboratórnych úloh sa pohybuje približne na úrovni CCNP (Cisco Certified Network Professional) čo sa približne vyrovnáva magisterskému stupňu štúdia.

1 SIMULAČNÉ PROSTREDIE GNS3

V tejto kapitole sa nachádza stručný prehľad GNS3 a dôležité súčasti tohto simulačného programu. Medzi ne patrí emulátor Dynamips a nástroje, ktoré slúžia na efektívne využitie výpočetných zdrojov. Tieto nástroje slúžia na spravovanie pamäťového a procesorového využitia. Spomínané súčasti sú popísané z dôvodu, aby bolo zrejmé, ako toto prostredie pracuje.

1.1 Prehľad GNS3

Pre simuláciu počítačových sietí existuje na trhu niekoľko programov. Jedným z nich je voľne dostupný GNS3. Toto prostredie ponúka širokú škálu emulácie sieťových prostriedkov, ako sú rôzne rady smerovačov a zariadenia zabezpečujúce bezpečnosť sietí. Prostredníctvom VMware alebo VirtualBox klienta sme schopní implementovať do našej simulovanej siete virtualizované stanice [10]. Využiť potenciál simulačného programu v praxi môžeme kvôli dvom prípadom [9]. V prvom prípade hovoríme či už o menšej alebo komplexnej topológii, ktorú chceme pred samotným reálnym nasadením otestovať. Pri tejto príležitosti môžeme testovať celú topológiu siete alebo jej časť. Ďalším prípadom, kedy je veľmi vhodné využiť simulačné prostredie sú edukačné účely. Toto môžu využiť napríklad študenti alebo jedinci, ktorí sa pripravujú na rôzne certifikačné skúšky týkajúce sa sieťových technológií.

Prostredie GNS3 je z pohľadu koncového užívateľa prívetivé vďaka prehľadnému grafickému rozhraniu. Na ľavej strane prostredia sa nachádza paleta s výberom zariadení, ktoré sú kategorizované podľa funkcií. V hornej časti prostredia sa nachádza lišta ponúk, ktorá mimo iného obsahuje najdôležitejšiu položku v rámci nastavení a tou je *Edit*. Súčasťou tejto ponuky je položka *Preferences*. Táto ponuka a prostredie GNS3 sú detailnejšie popísané v kapitole 7.1. Jadrom celého simulačného programu sú samotné emulátory, ktoré sú popísané v ďalšej časti.

1.2 Dynamips

V tabuľke 1.1 je zhrnutý zoznam a popis emulátorov, ktoré prostredie GNS3 používa. Ako je vidieť z popisu, Dynamips je schopný emulovať operačné systémy od firmy Cisco. Dynamips je kompatibilný s operačnými systémami Windows, Linux a Mac OS (Macintosh Operating System). Pôvodne bol navrhnutý na emuláciu smerovača Cisco 7200 [36]. Jeho funkcia je preložiť inštrukcie z daného IOS obrazu, ktoré sú pôvodne určené pre MIPS (Microprocessor without Interlocked Pipeline Stages)

procesory na inštrukcie, ktoré sú kompatibilné s procesormi Intel (Integrated Electronics) a AMD (Advanced Micro Devices), ktoré sú bežnou súčasťou klasických PC (Personal Computer).

K problematike tohto emulátoru treba podotknúť jeden dôležitý fakt. Dynamips nie je schopný emulovať prepínače [35]. Je to práve z toho dôvodu, že momentálne sa nedajú emulovať ASIC (Application Specific Integrated Circuit) procesory, ktoré používajú práve tieto zariadenia. Jedným z možných riešení je osadiť smerovač EtherSwitch modulom a využívať takto prispôsobené zariadenie ako prepínač. Treba si však uvedomiť, že takto modifikovaný smerovač nebude zastupovať plnohodnotný prepínač. Takéto riešenie má určité obmedzenia a niektoré funkcie nebudú podporované, ktoré by inak reálnym prepínačom boli. Konkrétny zoznam funkcií, ktoré nie je možné simulovať s EtherSwitch modulom je k dispozícií na webových stránkach GNS3 [12]. Ďalšou alternatívou, pokiaľ chceme emulovať prepínač, je využitie iného simulačného prostredie ako GNS3.

Ďalšou zaujímavosťou je fakt, že Dynamips nevie spracovať príkaz **reload** [36]. Riešenie ponúka reštart smerovača z kontextovej ponuky daného zariadenia. Po kliknutí na smerovač zvolíme funkciu *Reload*.

Tab. 1.1: Typy emulátorov v prostredí GNS3 [36].

Emulátor	**Emulované zariadenia**
Dynamips	Emulácia Cisco smerovačov.
Qemu	Emulácia ASA firewallov, Juniper a Vyatta smerovačov, Linuxových staníc.
Pemu	Istá variácia Qemu používaná v rámci PIX firewallov.
VirtualBox	Emulácia Juniper a Vyatta smerovačov, Linux a Windows staníc.

1.3 Výpočetné zdroje

Pokiaľ využívame GNS3 pre komplexnejšie topológie, resp. chceme použiť vo svojej topológii viacej zariadení, je vhodné oboznámiť sa so základnými nástrojmi pre efektívne využitie výpočetných zdrojov. Pre efektívne vynaloženie pamäte počítača, na ktorom je spustený GNS3 slúži *ghostios*, *mmap* a *sparemem* [27]. Funkcie *ghostios* a *sparsemen* spoliehajú práve na *mmap* a preto by mala byť funkcia *mmap* povolená. Pre efektívne využitie procesora slúži funkcia označovaná ako *idlepc*. Všetky uvedené funkcie sú zakomponované do prostredia GNS3. Bez týchto funkcií by sa fyzická aj virtuálna pamäť rýchlo zahltili a procesor by vykazoval vysoké využitie.

Ghostios je schopný výrazne znížiť využitie fyzickej pamäte počítača pokiaľ pracujeme so zariadeniami, ktoré využívajú rovnaký IOS [27]. Namiesto toho, aby si každý smerovač uchovával IOS vo svojej virtuálnej RAM (Random Access Memory) pamäti, počítač, na ktorom emulujeme zariadenia alokuje jednu zdielanú oblasť pamäte, ktorú budú tieto zariadenia používať. Ako príklad by sme mohli uviesť situáciu, kedy máme spustených 5 zariadení s rovnakým operačným systémom. Pokiaľ by jeden IOS obraz predstavoval veľkosť 50 MB, bez pomôcky ghostios by sme obsadili 250 MB pamäte. S pomocou ghostios obsadí IOS iba 50 MB pamäte.

Sparsemem znižuje množstvo virtuálnej pamäte používanej inštanciami smerovačov [27]. Toto je dôležité z pohľadu operačného systému, na ktorom pracuje GNS3. Napríklad 32 bitový operačný systém Windows limituje jeden proces na 2 GB virtuálnej pamäte. Na 32 bitovom operačnom systéme Linux sú to 3 GB virtuálnej pamäte. Povolením tejto funkcie alokujeme iba virtuálnu pamäť, ktorá je aktuálne používaná operačným systémom smerovača v danej inštancii a nie celú nakonfigurovanú fyzickú pamäť. Základnou myšlienkou tejto pomôcky je šetriť virtuálnu pamäť.

Idlepc funkcia slúži na výpočet tzv. idlepc hodnôt smerovačov. V kapitole 1.2 na strane 17 spomínaný Dynamips nevie určiť kedy smerovač vykonáva efektívnu činnosť a kedy sa nachádza v procese nečinnej slučky [27]. Táto nečinná slučka predstavuje dobu čakania na určitú akciu [36]. Tou je napríklad spracovanie paketu, výpis do konzoly, prepočet smerovacej tabuľky a iné. Tým pádom sa procesor hostiteľského počítača plne vyťaží. Idlepc zaháji analýzu na spustenom smerovači a stanoví najpravdepodobnejšie body v kóde, ktoré reprezentujú nečinnú slučku v rámci inštrukcií operačného systému IOS. Po tom, čo je hodnota zistená a aplikovaná na daný smerovač, Dynamips uvedie smerovač do režimu spánku, keď sa bude táto slučka vykonávať. Toto výrazne redukuje výpočetné procesy na počítači, zatiaľ čo smerovač pracuje bez akýchkoľvek obmedzení výkonu.

Idlepc hodnoty sa viažu na konkrétny IOS [27] a sú reprezentované ako hexadecimálna hodnota. Hodnoty sa budú líšiť pre rozdielne verzie IOS obrazov a to aj pokiaľ sa bude jednať o totožnú verziu s rozdielnym balíkom funkcií. Problematike IOS a jeho balíčkov sa venuje kapitola 2.3 na strane 22. Môže nastať prípad, kedy idlepc nedokáže nájsť správnu hodnotu pre IOS. Avšak pri opätovnom spustení idlepc procesu sa vypočítajú správne hodnoty. Po procese vyhľadávania sa vygeneruje niekoľko hodnôt, pričom pri niektorých sa bude nachádzať znak hviezdičky. Tieto označené hodnoty poukazujú na potenciálne najlepšie nájdené výsledky.

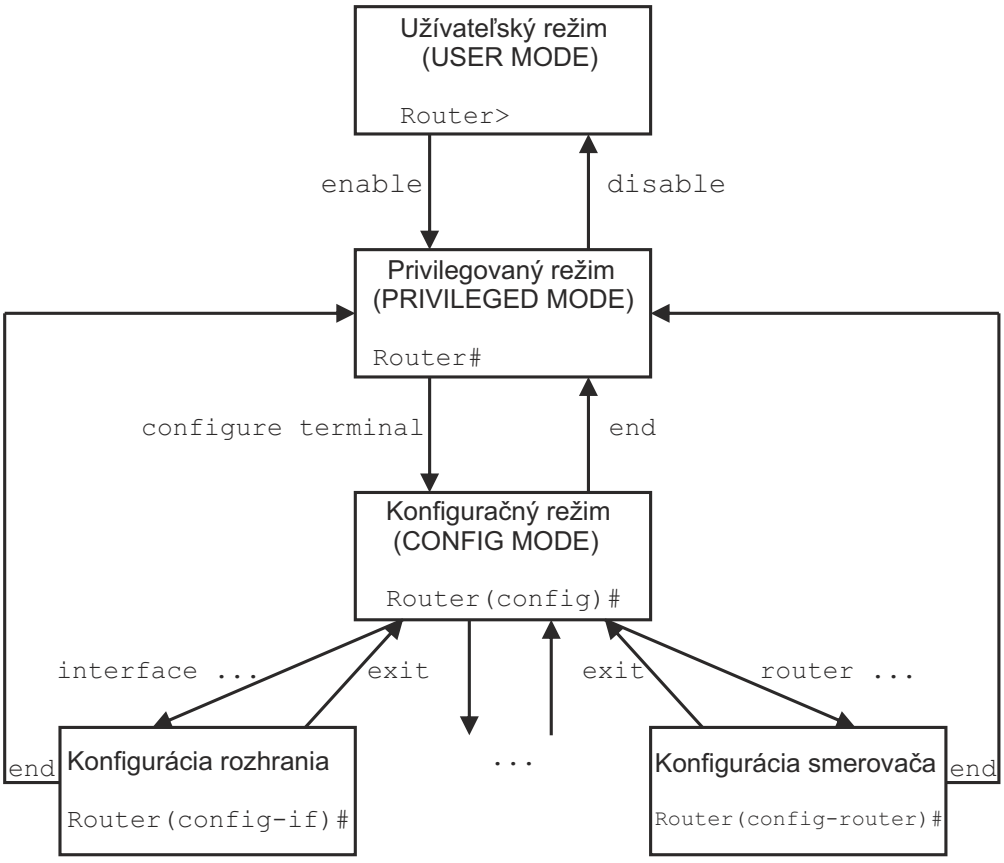
Mmap funkcia vynucuje, aby Dynamips využíval stránkovaciu pamäť oproti fyzickej pamäti pre inštancie smerovačov.

2 CISCO IOS

Operačný systém IOS je software používaný spoločnosťou Cisco [37]. Tento systém je možné nájsť v smerovačoch a po novom aj v prepínačoch. V minulosti prepínače využívali operačný systém s názvom CatOS (Catalyst Operating System). Catalyst je označenie rady prepínačov od firmy Cisco. Kedysi boli prepínače striktne radené ako zariadenia druhej vrstvy sieťového modelu a nemali funkciu smerovania. V posledných rokoch sa využíva IOS aj v prepínačoch z dôvodu, že moderné prepínače disponujú niektorými schopnosťami smerovačov. IOS si môžeme predstaviť ako balík smerovacích, prepínacích a telekomunikačných funkcií.

2.1 Rozhranie IOS

Rozhranie pomocou ktorého sme schopný prevádzať konfiguráciu zariadení sa nazýva CLI (Command Line Interface). Toto prostredie je rozdelené do rôznych režimov, v ktorých je rôzna sada príkazov prístupná v rámci konkrétnych používateľských práv daného užívateľa. Na obrázku 2.1 vidíme model základných režimov IOS.



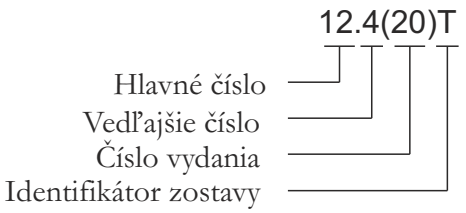
Obr. 2.1: Model režimov IOS [19].

2.2 Verzia IOS

Verzie operačných systémov sú značené číslicami a niekoľkými písmenami, ktoré majú špecifický význam [37]. Pre prehľadnosť a ilustráciu jednotlivých častí je priložený obrázok 2.2. Formát tohto značenia je **a.b(c.d)e**, pričom jednotlivé časti sú charakterizované v tabuľke 2.1:

Tab. 2.1: Význam symboliky označovania verzií IOS [37].

Symbol	Význam
a	Hlavné číslo verzie.
b	Vedľajšie číslo verzie.
c	Číslo vydania, ktoré začína číslom 1 a navyšuje sa každým vydaním novej aktualizácie v rámci rovnakej verzie **a.b**. Napríklad máme verziu 12.4(18), kde 18 sa navýši o jedna. Tak dostávame 12.4(19).
d	Priebežné číslo zostavy (vynechané zo všeobecných vydaní).
e	Identifikátor zostavy tzv. train. Cisco tento pojem vo svojej terminológii myslí ako vozidlo, ktoré poskytuje software ku konkrétnemu súboru platforiem a funkcií.



Obr. 2.2: Popis indentifikácie verzie IOS [37].

Trains, alebo tiež vozidlá ako je definované vyššie, sú vytvorené pre konkrétnych zákazníkov, ktorým Cisco poskytuje svoje služby [37]. Existuje niekoľko trainov. Výber niektorých trainov je definovaný v tabuľke 2.2. Od verzie IOS 15.0 a vyššie nie sú trainy T a M chápané ako samostatné trainy. Napríklad, prvé vydanie verzie 15.0 je označené ako 15.0(1)M, kde M indikuje rozšírenú podporu vydania. Táto rozšírená verzia pre údržbu vydaní poskytuje 44 mesačnú podporu. Verzia T bude označovať štandardnú verziu pre údržbu a poskytuje len 18 mesačnú podporu.

Tab. 2.2: Výber trainov a ich základný popis [37].

Značenie	**Význam**
Mainline	Tiež to môžeme chápať ako základnú verziu. Je charakterizovaný ako najviac stabilná verzia. Jeho sada funkcií sa nikdy nerozširuje. Predošlá verzia trainu sa stáva zdrojom aktuálnej verzie pre nový mainline. Napríklad train 12.1T sa stáva základom pre 12.2 mainline.
T	Technology train. Počas jeho životnosti je obohacovaný o rôzne záplaty a tým pádom je menej stabilný a bezpečný. Verzia sa neodporúča v rámci ostrej produkcie.
S	Service provider train. Táto verzia je upravená pre jadro (alebo tiež tzv. chrbticu) spoločnosti a je určená pre poskytovateľov služieb.
E	Enterprise train. Táto verzia je upravená pre podnikové siete.
B	Broadband train. Táto verzia je upravená pre medziľahlé siete, ktorých vlastnosťou je vysoká rýchlosť a priepustnosť dát.

2.3 Balíky IOS

Ďalšia kategorizácia spadá pod služby rôznych balíčkov. Balíček si môžeme predstaviť ako sadu špecifických funkcií, ktoré zastrešujú určitú oblasť sieťových technológií.

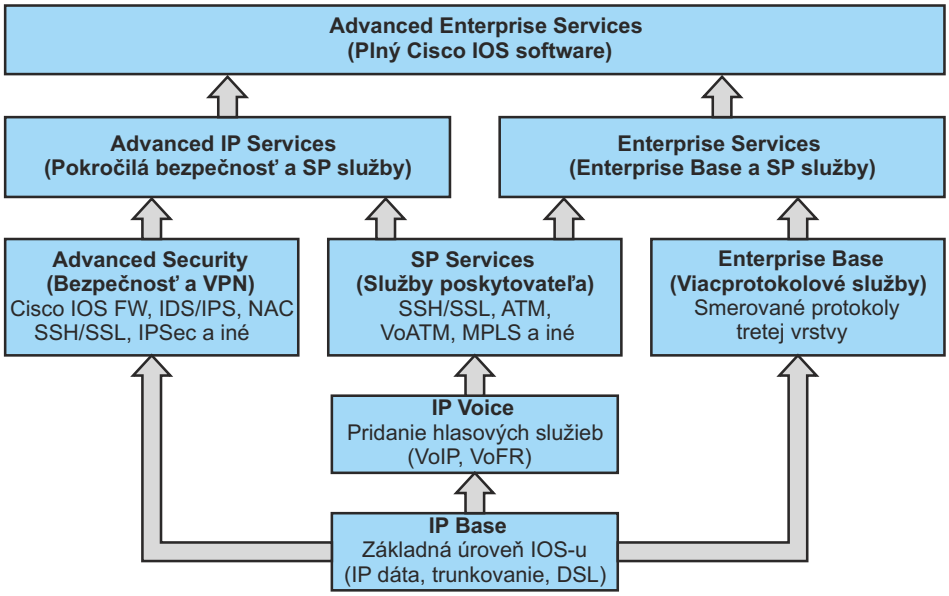
Tab. 2.3: Zoznam balíčkov pre Cisco smerovače a prepínače.

Smerovače [4]	**Prepínače** [37]
IP Base	Layer 2 Base
IP Voice	LAN (Local Area Network) Base
SP Services (Service Provider)	IP Base
Advanced Security	IP Services
Enterprise Base	Advanced IP Services
Enterprise Services	Enterprise Services
Advanced IP Services	Advanced Enterprise Services
Advanced Enterprise Services	

Hlavnú myšlienku zavedenia problematiky balíčkov sa dá zhrnúť nasledovne [4]:

- Zjednodušuje proces výberu správneho softwaru.
- Znižuje počet sád funkcií a tým znižuje zbytočnú komplexnosť pre danú kategóriu zákazníka.
- Konzistentné pomenovanie v rámci platforiem a nástrojov.

Obrázok 2.3 ilustruje hierarchickú pyramídu, ktorá zobrazuje náväznosť jednotlivých balíčkov. Pokročilejší balíček dedí funkcie z nižšieho balíčku. Zákazník, ktorý aktualizuje svoje zariadenie na vyššiu verziu balíčku, si zároveň ponecháva všetky funkcie z nižších [4].



Obr. 2.3: Hierarchická schéma IOS balíčkov [4].

- IP Base – Nezahŕňa kryptografické služby. Klasická IP dátová komunikácia, trunkovanie a DSL (Digital Subscriber Line).
- IP Voice – Nezahŕňa kryptografické služby. Zahŕňa VoIP (Voice over IP), VoFR (Voice over Frame Relay).
- SP Services – Zahŕňa SSH/SSL (Secure Shell/Secure Sockets Layer), ATM (Asynchronous Transfer Mode), VoATM (Voice over Asynchronous Transfer Mode), MPLS a iné.
- Advanced Security – Zahŕňa Cisco IOS FW (Firewall), IDS/IDP (Intrusion Detection System/Intrusion Detection and Prevention), NAC (Network Admission Control), SSH/SSL, IPsec (Internet Protocol Security) a iné.
- Enterprise Base – Nezahŕňa kryptografické služby. Zahŕňa smerované protokoly tretej vrstvy ako IPX (Internetwork Packet Exchange) a iné. Podpora *IP Base* pre IBM (International Business Machines).
- Enterprise Services – Nezahŕňa kryptografické služby. Plná podpora služieb pre IBM. Zlúčenie *SP Services* a *Enterprise Base*.
- Advanced IP Services – podpora IPv6, komplexnejšia bezpečnosť pre *SP Services*. Spája *Advanced Security* a *SP Servies*.
- Advanced Enterprise Services – Plná verzia Cisco operačného systému. Zlučuje *Advanced IP Services* a *Enterprise Services*

3 INTERNETOVÝ PROTOKOL IPV4

Táto kapitola stručne pojednáva o protokolovej sade IPv4. Zahrňuje adresáciu, popis štruktúry paketu, protokol ICMP a protokol DHCP.

3.1 Základný popis protokolu IPv4

Protokol IPv4 je dátovo orientovaný nespojový komunikačný protokol sieťovej vrstvy [25]. Sám o sebe nenadväzuje spojenie medzi komunikujúcimi uzlami. Pracuje na princípe *best effort*, čo vo voľnom preklade znamená vynaloženie najväčšej snahy o doručenie správ. Zaraďuje sa medzi nespoľahlivé protokoly, pretože negarantuje doručenie správ. O toto sa starajú protokoly vyšších vrstiev. V posledných rokoch nastáva problém s vyčerpaním adresného priestoru, ktorý protokol IPv4 poskytuje a aj to je jeden z mnohých dôvodov, prečo sa postupne nahradzuje protokolom IPv6.

3.2 Adresovanie

Protokolová sada IPv4 používa 32 bitové adresy [25]. Teoretický počet adries, ktoré môžeme využiť v rámci tohto protokolu je približne 4,2 miliardy. Najčastejšie sa stretneme s tzv. bodkovo decimálnym zápisom, ako udáva anglická literatúra. Ukážka tohto zápisu môže byť napríklad 192.168.1.0/24. Adresu reprezentujú štyri bloky po ôsmich bytoch. Tieto bloky sú tak isto nazývané oktety. Za IPv4 adresou sa nachádza údaj, ktorý identifikuje masku siete.

Pôvodne bola adresa delená na dve časti [21]. Časť najviac významných bitov reprezentovala sieť, zatiaľ čo menej významné bity reprezentovali koncové stanice. Takýto spôsob adresovania dovoľoval vytvoriť len 256 sietí. Toto prvotné delenie sa začalo javiť ako neefektívne a z toho dôvodu sa vytvorilo nové delenie, ktoré poznáme pod pojmom *classful*. Tabuľka 3.1 popisuje princíp adresného rozsahu.

Tab. 3.1: Classful rozdelenie adresného rozsahu [25].

Trieda	**1. Oktet**	**Maska siete**	**Počet možných sietí**
A	0–127	255.0.0.0	128
B	128–191	255.255.0.0	16 383
C	192–223	255.25.255.0	2 097 150
D	224–239	240.0.0.0	Skupinové adresy
E	240–255	240.0.0.0	Experimentálne adresy

Postupom času sa vytvorili nové techniky na efektívnejšie rozdelenie tohto adresného priestoru. Jedna z techník mala názov *variable–length subnet mask*, čo v preklade znamená variabilná dĺžka masky siete. Neskôr bola táto technika nahradená ďalšou technikou, ktorá je označovaná ako CIDR (Classless Inter–Domain Routing).

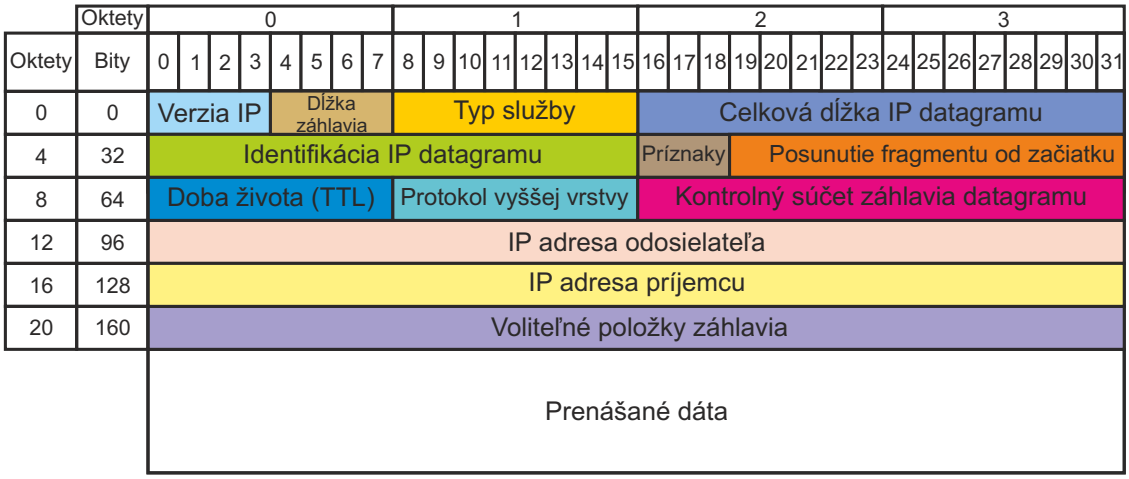
Ďalšie rozdelenie IPv4 adries je na základe vnútornej a vonkajšej siete. Tabuľka 3.2 popisuje rozdelenie privátnych adries, ktoré je možné používať v rámci vnútornej domácej alebo podnikovej siete. V rámci vonkajšej siete, akou je internet, tieto adresy nie sú smerované.

Tab. 3.2: Prehľad privátnych adries [25].

Adresný rozsah	**Najväčší CIDR blok**	**Počet adries**
10.0.0.0–10.255.255.255	10.0.0.0/8	16 777 216
172.16.0.0–172.31.255.255	172.16.0.0/12	1 048 576
192.168.0.0–192.168.255.255	192.168.0.0/16	65 536

3.3 Štruktúra paketu

Štruktúra hlavičky IP paketu je ilustrovaná obrázkom 3.1. Z obrázka je vidieť, že povinná hlavička má veľkosť 20 bajtov (alebo tiež oktetov). V prípade potreby je možné využiť pole *Voliteľné položky záhlavia*, ktoré navýši veľkosť hlavičky na 24 bajtov. V nasledujúcej sekcií je stručný popis jednotlivých polí záhlavia IP paketu. Čerpané bolo zo zdrojov [24] a [25].



Obr. 3.1: Hlavička protokolu IPv4 [25].

Verzia IP (4 b) – Toto pole identifikuje verziu IP protokolu. V tomto prípade obsahuje hodnotu 4.

Dĺžka záhlavia (4 b) – Pole informuje o veľkosti záhlavia. Je dôležité práve z dôvodu, že záhlavie môže mať premennú dĺžku a to 20 až 60 bajtov v násobkoch 32 bitov čo sú 4 bajty.

Typ služby (8 b) – Toto 8 bitové pole definuje prioritu paketu. Využíva sa v rámci služby QoS (Quality of Service).

Celková dĺžka IP datagramu (16 b) – Nesie informáciu o celkovej dĺžke paketu, pričom je zahrnutá aj veľkosť hlavičky paketu. Maximálna veľkosť je 65 536 bajtov.

Identifikácia IP datagramu (16 b) – Unikátna hodnota, ktorá má identifikovať IP paket v rámci viacerých fragmentov.

Príznaky (3 b) – Toto pole je reprezentované tromi bitmi, ale používajú sa len dva. Jeden je totiž rezervovaný. Pole nesie informácie o fragmentácii paketov. Ak je DF–bit (Don't Fragment) nastavený na hodnotu 1, fragmentácia paketov je zakázaná. Ak MF–bit (More Fragments) má hodnotu 1, nasleduje ďalšia časť fragmentovaného datagramu.

Posunutie fragmentu od začiatku (13 b) – Určuje pozíciu dát datagramu v rámci začiatku pôvodného paketu. Vďaka tomuto polu je strana prijímateľa schopná fragmentovaný paket znovu poskladať.

Doba života (TTL) (8 b) – Pole je nastavené na strane odosielateľa paketu, pričom sa hodnota dekrementuje o 1 na každom smerovači, ktorým paket prejde. Inými slovami môže označovať aj počet skokov v sieti v rámci smerovačov. Paket sa zahadzuje, pokiaľ pole nadobudne hodnotu nula. Platný rozsah tohto pola je 1–255.

Protokol vyššej vrstvy (8 b) – Označuje použitý protokol, v rámci ktorého je paket zasielaný. Napríklad TCP (Transmission Control Protocol) má hodnotu 6, UDP (User Datagram Protocol) hodnotu 17 a ICMP hodnotu 1.

Kontrolný súčet záhlavia datagramu (16 b) – Cyklická redundantná kontrola vykoná kontrolu samotného záhlavia. Keďže hodnota TTL sa spracovaním každého smerovača dekrementuje, kontrola je vykonaná na každom smerovači v rámci trasy paketu. Ak sa hodnota súčtu nerovná paket je zahodení.

IP adresa odosielateľa (32 b) – Pole uchováva IP adresu odosielateľa paketu.

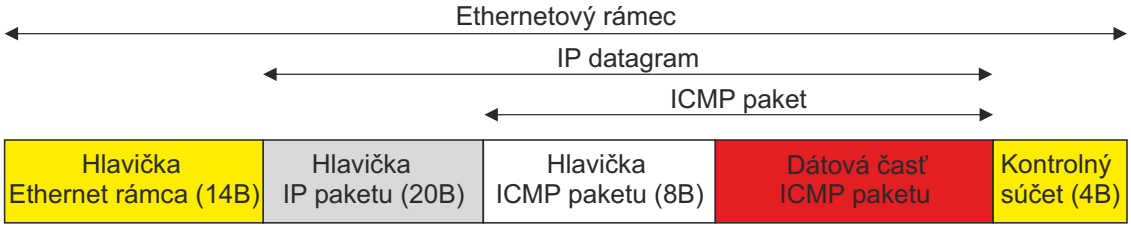
IP adresa príjemcu (32 b) – Pole uchováva IP adresu príjemcu paketu.

Voliteľné položky záhlavia (32 b) – Toto pole sa používa minimálne. Niektoré položky sú dokonca zakázané. Využitie si nájde napríklad v rámci testovania siete.

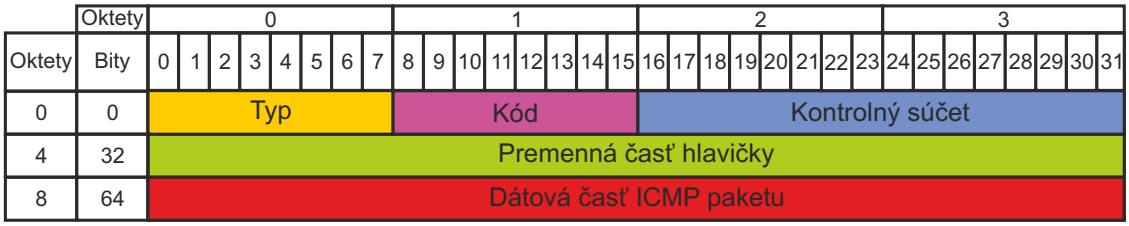
3.4 Protokol ICMP

Internetový protokol IP nie je spoľahlivý a sám o sebe nenadväzuje konektivitu medzi komunikujúcimi uzlami [32]. V rámci tretej vrstvy sieťového modelu ISO/OSI (International Organization for Standardization Open Systems Interconnection) spadá aj protokol ICMP. Nie je priamou súčasťou protokolovej sady IPv4, ale riadi určite aspekty IPv4 prevádzky a poskytuje diagnostické mechanizmy. Slúži na ohlasovanie rôznych správ ohľadne sieťovej komunikácie. Protokol ICMP by mal byť implementovaný na každom zariadení pracujúcom na tretej vrstve.

ICMP správy sú zasielané s použitím základnej IP hlavičky, ako ilustruje obrázok 3.2. V tejto hlavičke sa bude nachádzať pole *Protokol vyššej vrstvy* s hodnotu 1, ktorá reprezentuje ICMP. Z tohto vyplýva, že ICMP správy nevyužívajú k svojmu prenosu vyššie protokoly ako TCP, UDP alebo SCTP (Stream Control Transmission Protocol) [15]. Samotná ICMP hlavička je znázornená na obrázku 3.3. Prvý oktet ICMP hlavičky označuje *Typ* správy. Hodnota tohto pola určuje formát nasledujúcich častí ICMP správy. Hodnota pola *Kód* slúži na bližšiu špecifikáciu typu správy. Výber niektorých typov a kódov je obsiahnutý v tabuľke 3.3. Hodnota pola *Kontrolný súčet* sa počíta na základe ICMP hlavičky a dátovej časti. Premenná časť hlavičky je závislá na konkrétnom type ICMP správy. Bližšia špecifikácia týchto správ je obsiahnutá v dokumente RFC 792 [32].



Obr. 3.2: Zapúzdrenie ICMP paketu [20].



Obr. 3.3: Hlavička protokolu ICMP [32].

Tab. 3.3: Výber niektorých typov ICMP správ a ich význam [17].

Typ	**Kód**	**Význam správy**
0 – Echo reply	0	Echo odpoveď (pri použití ping).
1,2	–	Rezervované.
3 – Destination unreachable	0	Nedostupnosť cieľovej siete.
	1	Nedostupnosť koncovej stanice.
	4	Požadovaná fragmentácia.
	6	Neznáma cieľová stanica.
4 – Source quench	0	Zníženie rýchlosti toku dát.
5 – Redirect message	0	Presmerovanie datagramu pre sieť.
	1	Presmerovanie datagramu koncovej stanici.
8 – Echo request	0	Echo požiadavka (pri použití ping).
11 – Time exceeded	0	Hodnota TTL nadobudla nulovú hodnotu.

3.5 Protokol DHCP

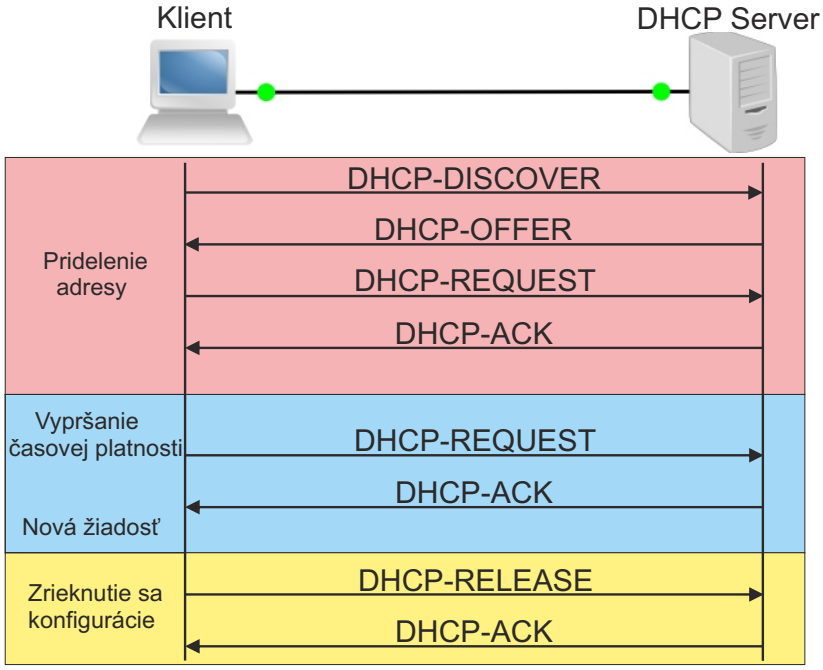
Aplikačný protokol DHCP pracuje na báze klient server [25]. Slúži na dynamickú konfiguráciu adries jednotlivých staníc v sieti. V problematike DHCP figurujú tri entity a sú to server, klient a relay agent. Vďaka tomuto protokolu je stanica schopná z DHCP servera obdržať parametre, akými sú napríklad [2] [25]:

- IP adresa a maska siete
- Predvolená brána
- DNS (Domain Name System) Servery
- WINS (Windows Internet Naming Service)
- Názov domény
- Názov hostiteľa
- Časový server
- Adresa TFTP serveru

Využívanie DHCP protokolu ponúka niekoľko výhod:

- Odpadá nutnosť konfigurácie každého jednotlivého klienta v sieti.
- Nie je potrebné držať záznamy o pridelenom adresnom rozsahu.
- Pridelenie novej IP adresy v prípade presunutia stanice do inej siete.
- Uvoľnenie IP adresy koncovej stanice pokiaľ je v režime offline (šetrenie adresného priestoru).
- Duplicitné pridelenie IP adries je prakticky nemožné, keďže proces pridelenia je plne automatizovaný.

Princíp fungovania DHCP protokolu je ilustrovaný na obrázku 3.4. Ako prvé vyšle koncová stanica správu *DISCOVER*, ktorú zašle na všesmerovú adresu, pričom zdrojová IP adresa je 0.0.0.0 [31]. Jeden alebo aj niekoľko DHCP serverov zachytí túto správu a následne vyšle správu typu *OFFER*. Koncová stanica odpovie prvému DHCP serveru správou *REQUEST*. V tomto prípade sa ešte stále jedná o všesmerovú adresu. DHCP server pošle stanici správu typu *ACK* s údajmi konečnej konfigurácie. V praxi sa pridelia adresné parametre len na určitú dobu. Po uplynutí polovice tejto doby si koncová stanica vyžiada novú konfiguráciu. Pokiaľ stanica nevyšle žiadosť, DHCP server nesmie prideliť stanici novú konfiguráciu. Pokiaľ sa stanica vzdáva svojej konfigurácie vyšle správu *RELEASE*. Tabuľka 3.4 obsahuje popis jednotlivých typov správ v rámci DHCP komunikácie.



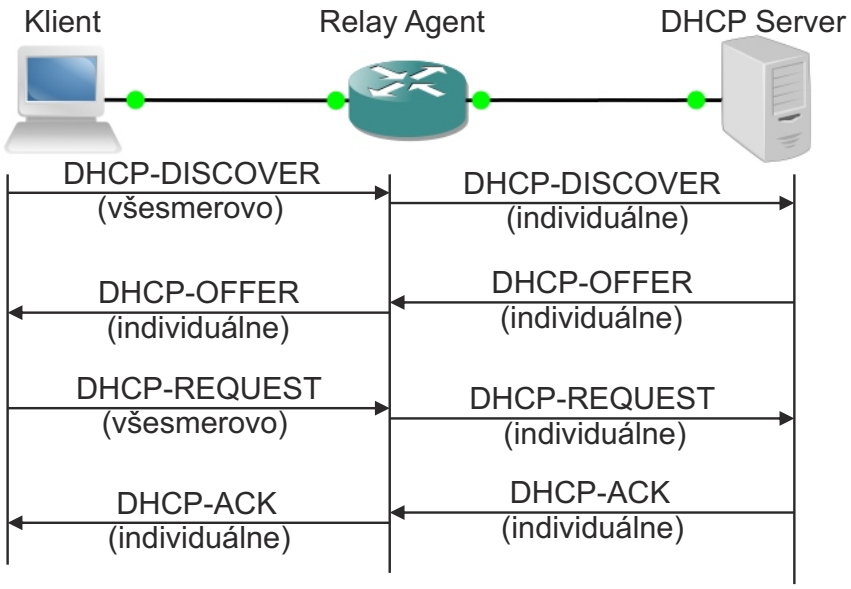
Obr. 3.4: Základný princíp DHCP protokolu [31].

V bežnej praxi je častým prípadom situácia, keď sa DHCP server nenachádza v rovnakej sieti ako samotný klienti. Riešenie ponúka relay agent, ktorého princíp

ilustruje obrázok 3.5 [24]. Na smerovači je konfigurácia, ktorá zabezpečí to, aby správy od koncových staníc boli presmerované na DHCP server, ktorý sa nachádza v inej sieti. Relay agent odošle tieto správy DHCP serveru s informáciou o zdrojovej sieti. Takto DHCP server vie prideliť parametre zo správneho rozsahu adries.

Tab. 3.4: Typy DHCP správ a ich význam [31].

Správa	**Význam**
discover	Koncová stanica hľadá DHCP serveri. Jedná sa o UDP všesmerovú správu. Zdrojový port 68, cieľový port 67.
offer	Odpoveď DHCP servera na správu DHCP discover. Jedná sa o UDP všesmerovú správu. Zdrojový port 67, cieľový port 68.
request	Žiadosť koncovej stanice pre konkrétny DHCP server.
decline	Komunikácia klient–server. Indikuje, že IP adresa sa už používa.
ack	Komunikácia server–klient. Server odpovedá na správu request. Táto správa obsahuje všetky konfiguračné parametre.
nack	Komunikácia server–klient. Server odpovedá na správu request. V tomto prípade ale server zamietol požiadavku klienta.
release	Komunikácia klient–server. Správa indikuje, že koncová stanica sa zriekla svojej konfigurácie.
inform	Komunikácia klient–server. Používa sa v prípade, keď klient žiada o dodatočné parametre.



Obr. 3.5: Princíp DHCP protokolu s relay agentom [14].

4 INTERNETOVÝ PROTOKOL IPV6

Táto kapitola stručne pojednáva o protokolovej sade IPv6. Obsahuje základný popis tohto protokolu, adresáciu, popis štruktúry paketu, protokol ICMPv6, DHCPv6, migračné techniky v rámci IPv6 a IPv4 a na záver je stručne uvedená problematika bezpečnosti protokolu IPv6. V rámci bezpečnosti sú načrtnuté oblasti, ktoré sa týkajú priamo samotných laboratórnych úloh.

4.1 Základný popis protokolu IPv6

Protokol IPv6 je dátovo orientovaný komunikačný protokol spadajúci do tretej tzv. sieťovej vrstvy sieťového modelu ISO/OSI. Hlavným dôvodom vývoja a nasadenia tohto protokolu je nedostatok adresného priestoru jeho predchodcu IPv4 a iné bezpečnostné dôvody [25]. Technológia NAT (Network Address Translation) umožnila do istej miery konkurencie schopnosť protokolu IPv4, avšak problém nevyriešila [24]. Navyše využitím technológie NAT sa degradovala hlavná myšlienka internetu a tou je tzv. end–to–end komunikácia.

Protokol IPv6 nebol vytvorený len kvôli navýšeniu adresného priestoru. Nová protokolová sada má niekoľko výhod oproti predošlej verzii, kvôli ktorým je vhodné ju nasadiť. V nasledujúcich riadkoch je obsiahnutý stručný popis výhod a inovácií. Čerpané bolo zo zdrojov [24] a [30].

Funkcie pridelenia adries: IPv6 podporuje niekoľko metód pre dynamické pridelenie adries vrátane DHCP a bezstavovej autokonfigurácie.

Kvalita služieb: Mechanizmy pre priame zaistenie QoS. Tok dát sa značkuje v záhlaví paketu s využitím poľa s nazvom *Identifikácia toku dát*.

Vstavaná podpora mobility: Koncové stanice sa môžu pohybovať mimo lokálnu sieť, pričom si zachovávajú IPv6 adresu bez straty aktuálnej relácie.

Dĺžka paketu: Protokol IPv6 dokáže pracovať s paketmi o veľkosti až 4 GB. Tieto pakety sa nazývajú jumbo pakety a slúžia na špeciálne účely.

Problematika NAT: Veľký adresný priestor IPv6 umožňuje vypustenie NAT techniky, ktorá v niektorých prípadoch spôsobovala aplikáciam ťažkosti. Vylúčenie techniky NAT zefektívňuje smerovanie dát v sieti.

Multicast: Lepšia podpora multicastovej prevádzky.

Poskytovanie závislých a nezávislých verejných adries: Poskytovatelia internetového pripojenia môžu prideliť verejný IPv6 adresný priestor (závislý) alebo spoločnosti môžu registrovať ich vlastný verejný adresný priestor (nezávislý).

Veľkosť adresy: Dĺžka IPv6 adresy je 128 bitov čo predstavuje veľmi ťažko vyčerpateľný adresný priestor.

Agregácia: Veľký adresný priestor vytvára lepšie podmienky na rozumnejšie zoskupenie adresných blokov v rámci internetu. Verejné IPv6 adresy sú zoskupené do hlavných geografických oblastí. V rámci každej oblasti je adresný priestor ďalej rozdelený poskytovateľom služieb. V rámci každej oblasti poskytovateľa je adresný priestor ďalej rozdelený pre jednotlivých zákazníkov. Z toho vyplýva efektívnejšie smerovanie.

IPsec: V IPv4 použitie tejto techniky nebolo povinné, avšak IPv6 vyžaduje aby každá implementácia podporovala IPsec. Nevyžaduje aby každé zariadenie používalo IPsec, ale akékoľvek zariadenie, ktoré implementuje IPv6 musí mať schopnosť implementovať IPsec.

Vylepšenia hlavičky: V rámci IPv6 hlavičky je niekoľko vylepšení v porovnaní s IPv4. Napríklad smerovače neprepočítavajú kontrolný súčet pre každý paket. Hlavička obsahuje nové pole s názvom identifikácia toku dát, ktoré slúži na jednoduchú identifikáciu paketov zaslaných cez rovnaké TCP alebo UDP spojenie. Hlavička IPv6 je jednoduchšia vďaka absencii niektorých položiek z predošlej verzie IPv4.

Absencia všesmerových adries: IPv6 narozdiel od IPv4 nepoužíva všesmerové adresy. Procesy využívajúce všesmerovú adresu sú v rámci IPv6 riešené pomocou skupinových adries.

4.2 Adresovanie

Jednou z hlavných inovácií, ako bolo spomenuté v predošlej kapitole, je samotná dĺžka prenášanej adresy. Nová protokolová sada IPv6 disponuje so 128 bitovou adresou, zatiaľ čo predošlá verzia mala 32 bitovú. Je to veľký skok, pretože týmto sa zvýšil počet možných adries z približne 4,2 miliárd až na 340 sextiliónov.

Formát adresy je reprezentovaný ako zoskupenie ôsmich 16 bitových polí [34]. Každé pole je zapísané ako hexadecimálna hodnota v rozsahu 0x0000 až 0xFFFF oddelené dvojbodkou. Jednotlivé znaky A,B,C,D,E a F sa môžu písať malým alebo

veľkým písmom. Zápis IPv6 adresy nemá striktnú notáciu a existujú určité pravidlá, ktoré zjednodušujú zápis už tak dlhej adresy.

Pravidlá zápisu IPv6 adresy [30]:

- Adresu je možné písať v jednoduchšej forme vynechaním núl pričom je pravidlo, že sa vynechávajú nuly na začiatku poľa. Napríklad segment :07F0: môžeme písať ako :7F0:.
- Skupina núl v segmente sa môže písať ako samotná nula. Napríklad segment :0000: zapíšeme ako :0:.
- Zoskupenie niekoľkých polí, ktoré obsahujú samé nuly môžeme písať ako ::. Takýto zápis môžeme použiť len raz v celej adrese a to z jednoduchého dôvodu. Keby sme túto notáciu použili viackrát, neboli by sme schopní zistiť, v ktorej skupine bol aký počet núl. Tým pádom by sme nevedeli jednoznačne identifikovať adresu.

Protokol IPv6 umožňuje prideliť jednému fyzickému rozhraniu niekoľko IPv6 adries rôzneho typu. V rámci adries sa môžeme stretnúť s troma skupinami, ktoré sa delia na ďalšie kategórie [24]:

- Individuálne
  - Globálne unikátne
  - Linkové unikátne
  - Lokálna slučka
  - Nešpecifikovaná
  - Lokálna unikátna
  - IPv4 kompatibilné
- Výberové
- Skupinové

Bližšia špecifikácia rôznych typov IPv6 adries je popísaná v nasledujúcich riadkoch. Informačné zdroje, z ktorých bolo čerpané sú [22], [24] a [30].

Nešpecifikovaná ::/128 – Táto adresa sa používa ako zdrojová adresa koncovej stanice pred tým, ako jej bola pridelená vlastná adresa.

Lokálna slučka ::1/128 – Využitie tejto adresy je v rámci jednej sieťovej karty. Napríklad keď v rámci koncovej stanice komunikujú dve rôzne aplikácie. IPv4 ekvivalentom je 127.0.0.0/8. Rozdiel spočíva v tom, že v IPv4 sa jednalo o celý rozsah adries zatiaľ čo u IPv6 sa jedná o jednu konkrétnu adresu.

Teredo 2001:0000::/32 – Adresa umožňujúca IPv6 tunelovanie cez IPv4 NAT.

IPv4 mapované ::FFFF/96 – Tento typ adresy sa používa v rámci zapúzdrenia IPv4 adresy do adresy IPv6. Toto sa využíva napríklad v sieti typu dual stack. Tento typ siete je popisovaný v kapitole 4.6.

Lokálna unikátna FC00::/7 – Rezervované pre použitie v privátnych sieťach. Ich ekvivalentom v IPv4 sú adresy 10.0.0.0/8, 172.16.0.0/12 a 192.168.0.0/16.

Linková unikátna FE80::/10 – Tieto adresy sú používané v rámci jednej linky alebo v sieťach typu Ethernet. Nemusia byť unikátne mimo túto sieť. Tento typ adries nesmú smerovače smerovať. Ekvivalentom v IPv4 sú autokonfiguračné adresy 169.254.0.0/16.

Experimentálne 2001:0002::/48 – Tieto adresy sú rezervované pre použitie v dokumentáciách. Nemali by byť použité ako cieľové alebo zdrojové adresy.

Orchid 2001:0010::/28 – Tento typ adries je použitý na určitú dobu. Mali by byť viditeľné pre end–to–end komunikácie a smerovače by nemali vidieť pakety, v ktorých sú tieto adresy použité či ako zdrojové alebo cieľové adresy.

6to4 2002::/16 – Používajú sa pri nasadení techniky 6to4 (Connection of IPv6 Domains via IPv4 Clouds). Do druhého a tretieho kvartálu sa vpíše IPv4 adresa v hexadecimálnej forme. Táto tranzitná technika umožňuje IPv6 pakety zasielať cez IPv4 sieť bez vytvárania explicitných tunelov.

Dokumentačné 2001:db8::/32 – Adresy tohto typu sú používané ako ukážkové príklady v dokumentáciách. Nemali by byť nikdy použité ako cieľové alebo zdrojové. Ekvivalentom v IPv4 sú to adresy 192.0.2.0/24, 198.51.100.0/24 a 203.0.113.0/24.

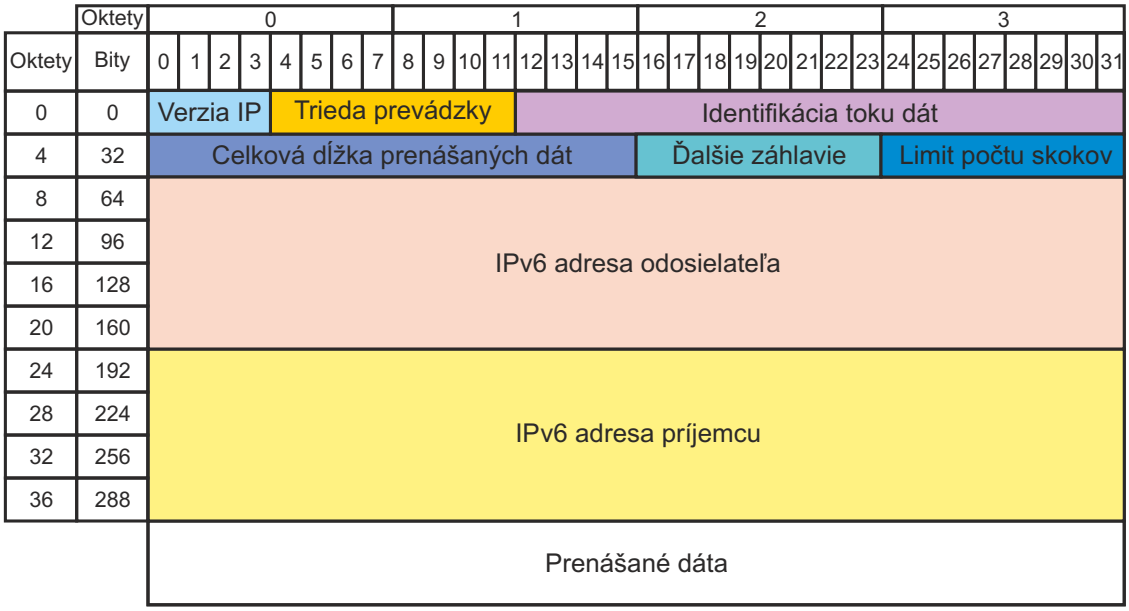
Globálne unikátne 2000::/3 – Adresy, ktoré sa používajú pre bežné použitie mimo privátnych sietí. V rámci IPv4 ich zastupujú verejné adresy.

Výberové – Tieto adresy identifikujú určitú skupinu staníc alebo zariadení. Jedna výberová adresa identifikuje niekoľko fyzických rozhraní. Paket zaslaný na túto adresu bude doručený najbližšiemu zariadeniu, ktoré zdieľa túto adresu. Najbližšie zariadenie sa určuje za pomoci použitého smerovacie protokolu.

Skupinové adresy FF00::/8 – Adresy sú používané pre identifikáciu skupiny v sieti. Sú používané len ako cieľové adresy. Ekvivalentom v IPv4 sú to 224.0.0.0/4.

4.3 Štruktúra paketu

Aj napriek tomu, že hlavička IPv6 paketu obsahuje menej povinných polí a zároveň je tým jednoduchšia, jej veľkosť sa oproti predošlej verzií dvojnásobne zväčšila. Celkovo má základná hlavička protokolu IPv6 veľkosť 40 oktetov, pričom základná hlavička IPv4 má 20 oktetov. V predošlých kapitolách bola uvedená veľkosť adresného pola príjemcu a odosielateľa ako suma 256 bitov. Samotná suma týchto častí hlavičky tak predstavuje štvornásobne väčšiu hodnotu, na rozdiel od verzie IPv4, kde to bolo len 64 bitov [6] [25]. Obrázok 4.1 ilustruje základnú hlavičku IPv6.



Obr. 4.1: Hlavička protokolu IPv6 [6].

Nasledujúce riadky popisujú jednotlivé položky záhlavia IPv6 paketu [6].
Verzia IP (4 b) – Toto pole udáva, o ktorú verziu IP protokolu sa jedná. V tomto prípade pole obsahuje hodnotu 6.

Trieda prevádzky (8 b) – Pole umožňuje nastavenie priority paketu. V predošlej verzii IP protokolu sa jednalo o položku ToS (Type of Service).

Identifikácia toku dát (20 b) – Úlohou tohto pola je označovanie toku dát. Pakety označené rovnakým číslom sú zasielané určitou cestou v sieti. Využitie pre viacvrstvové prepínanie a vyššiu výkonnosť v rámci paketového prepínania.

Celková dĺžka prenášaných dát (16 b) – Pole indikuje veľkosť samotných prenášaných dát. To znamená bez hlavičky protokolu. Hodnota je udávaná v bajtoch.

Ekvivalentom v rámci verzie IPv4 to bolo pole označované ako *Celková veľkosť IP datagramu*.

Ďalšie záhlavie (8 b) – Toto pole nesie informáciu o použitom protokole vyššej vrstvy ako napríklad TCP, UDP, SCTP a iné. Ďalším prípadom využitia je informácia o rozširujúcom záhlaví, ktoré nie je povinné, ale smerovače si v rámci komunikácie môžu vymieňať dodatočné informácie.

Limit počtu skokov (8 b) – V predošlej verzií protokolu IPv4 sa toto pole nazývalo TTL (Time to Live) tzv. *Doba života* paketu. V tomto prípade sa jedná o presnejšiu definíciu. V rámci prenosu paketov zo smerovača na smerovač sa táto hodnota dekrementuje o jedna. Hodnota udáva maximálny počet skokov v rámci cesty paketu.

IPv6 adresa odosielateľa (128 b) – Pole uchováva IPv6 adresu odosielateľa paketu.

IPv6 adresa príjemcu (128 b) – Pole uchováva IPv6 adresu príjemcu paketu.

4.4 Protokol ICMPv6

Protokol ICMP sa vyskytuje aj u protokolovej sady IPv6. V tomto prípade je však označovaný ako ICMPv6. Štruktúra hlavičky protokolu a význam jednotlivých polí sa nezmenil a je totožný s ICMP [5]. Táto hlavička je ilustrovaná obrázkom 3.3. Výber niektorých typov ICMPv6 správ je spísaný v tabuľke 4.1. Implementácia ICMPv6 je povinná na všetkých zariadeniach, ktoré implementujú IPv6. Protokol IPv6 by bol bez ICMPv6 nefunkčný. Jedným z podstatných dôvodov je aj fakt, že pred samotným vyslaním paketu stanica musí zistiť MTU (Maximum Transmission Unit). V sieti IPv6 nenastáva fragmentácia v rámci prenášania paketu medzi smerovačmi. Koncové stanice stanovia MTU hneď na začiatku vysielania dát. Hodnota MTU sa zistí práve vďaka ICMPv6 správ. ICMPv6 sa tak isto používa na ohlasovanie rôznych správ ohľadne sieťovej komunikácie a testovanie dostupnosti špecifického uzla. Navyše sa tento protokol využíva k objavovaniu susedov, k správe multicastových skupín, prekladu adries a zaistení mobility. Z predchádzajúceho popisu je zrejmé, že prevádzku ICMPv6 správ nie je vhodné blokovať. Pri filtrovaný správ treba byť obozretný a zvoliť podmienky filtrovania tak aby neohrozili správnu prevádzku siete. Riešenie taktiež ponúka autentizačná hlavička. V zabezpečených sieťach sa ICMPv6 pakety zahadzujú, pokiaľ nie sú opatrené autentizačnou hlavičkou.

Tab. 4.1: Výber niektorých typov ICMPv6 správ a ich význam [18].

Typ	Kód	Význam správy
1 – Destination Unreachable	0	Neexistujúca cesta k cieľu.
	3	Nedosiahnuteľnosť adresy.
2 – Packet Too Big	0	Príliš veľký paket (Problematika MTU).
3 – Time Exceeded	0	Vypršanie hodnoty doby života.
128 – Echo Request	0	Echo požiadavka (pri použití ping).
129 – Echo Reply	0	Echo odpoveď (pri použití ping).
133 – Router Solicitation	0	Výzva smerovaču.
134 – Router Advertisement	0	Oznámenie smerovača.
135 – Neighbor Solicitation	0	Výzva susedovi.
136 – Neighbor Advertisement	0	Oznámenie suseda.

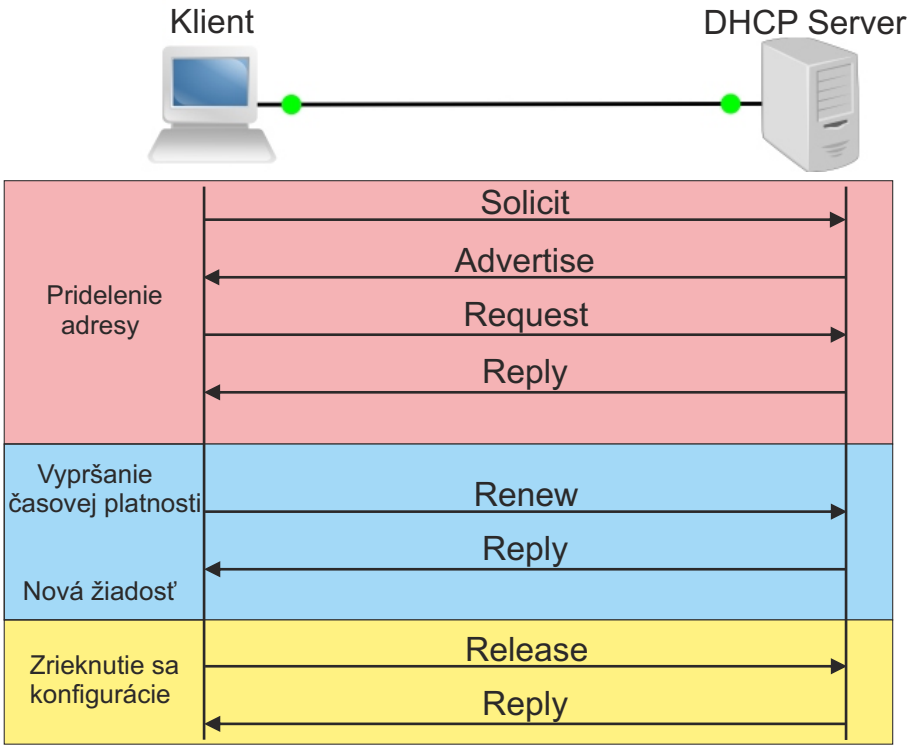
4.5 Protokol DHCPv6

Základná myšlienka protokolu DHCP pre IPv6 ostala rovnaká. Avšak DHCPv6 prináša so sebou isté zmeny. V prípade DHCPv6 sú servery a klienti jednoznačne identifikovaní na základne DUID (DHCP Unique IDentifier) [24]. V prípade DHCPv4 predstavovala jednoznačný identifikátor MAC adresa. V rámci jedného klienta je nutné identifikovať niekoľko rozhraní. Na toto slúži identifikátor IA (Identity Association). IA reprezentuje zoskupenie konfiguračných informácií pridelených danému rozhraniu, pričom je toto zoskupenie označené jednoznačným identifikátorom IAID (Identity Association Identification). Výber niektorých DHCPv6 správ obsahuje tabuľka 4.2. V rámci IPv6 hovoríme o stavovej a bezstavovej DHCPv6 konfigurácií.

Stavový DHCPv6 uchováva stav informácií ohľadne jednotlivých klientov [30]. Medzi tieto informácie patrí zapožičaná IPv6 adresa a dĺžka času zapožičania. Na tejto istej báze pracuje DHCPv4, ktorého popis nájdeme v kapitole 3.5 na strane 28. Obrázok 4.2 ilustruje princíp DHCPv6. Zmena nastala v spôsobe hľadania DHCPv6 serverov a názvoch niektorých správ. V tomto prípade sa správa *Solicit* odošle na skupinovú adresu FF02:1:2. Správa typu *Advertise* zahŕňa v sebe údaje ako IPv6 prefix, dĺžka prefixu a DNS adresy. Na rozdiel od DHCPv4 protokol DHCPv6 nedisponuje informáciami o predvolenej bráne. DHCPv6 sa spolieha na protokol NDP (Neighbor Discovery Protocol), v rámci koncovej stanice a miestnych smerovačov. Stavový DHCPv6 sa však v bežnej prevádzke nepoužíva.

Tab. 4.2: Typy DHCPv6 správ a ich význam [33].

Správa	**Význam**
solicit	Hľadanie DHCPv6 serverov.
advertise	Odpoveď DHCPv6 servera na správu solicit.
request	Žiadosť o konfiguráciu pre koncovú stanicu od DHCPv6 servera.
renew	Žiadosť koncovej stanice o predĺženie doby zapožičania konfigurácie. Správa sa posiela danému DHCPv6 serveru, od ktorého má stanica pôvodnú konfiguráciu.
rebind	Pokiaľ stanica nedostane odpoveď na správu renew, stanica odošle správu rebind inému dostupnému DHCPv6 serveru.
relay–forw	Správa pochádzajúca od relay agenta. Je cielená DHCPv6 serveru a jedná sa o preposlanie žiadostí koncových staníc.
relay–repl	Správa pochádzajúca od DHCPv6 serveru určená pre relay agenta ako odpoveď na správu relay–forw.
reply	Odpoveď DHCPv6 servera na správu solicit, request, renew, rebind.



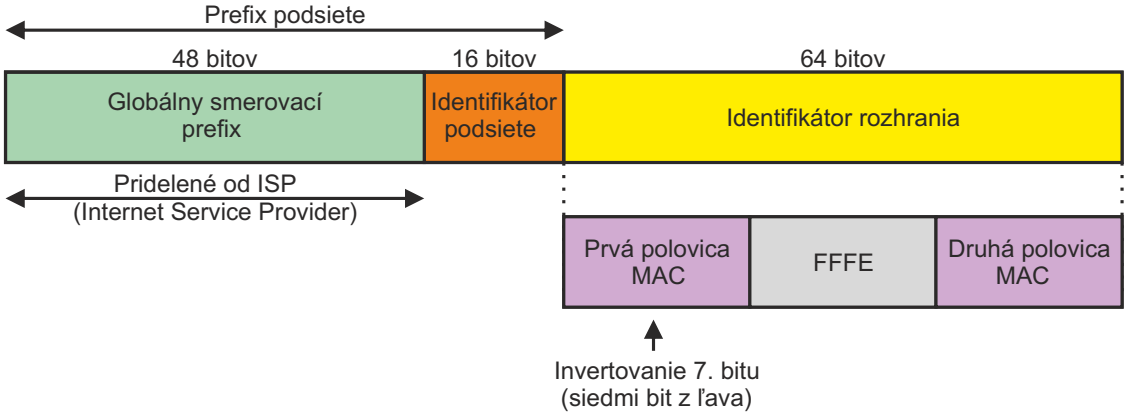
Obr. 4.2: Základný princíp DHCPv6 protokolu [7].

Bezstavový DHCPv6 sa tak isto radí medzi dynamické konfigurácie adresných parametrov koncových staníc [30]. Hlavný rozdiel oproti stavovej konfigurácií je v spôsobe získavania špecifických údajov. IPv6 prefix, dĺžku prefixu a predvolenú bránu si koncová stanica získa za pomoci NDP. Klientská časť adresy sa odvodí za pomoci techniky EUI–64 (Extended Unique Identifier). Bezstavový DHCPv6 sa využije na získanie DNS IPv6 adries.

Koncová stanica vyšle skupinovú správu, ktorú zachytia všetky smerovače na danej sieti. Tento proces prebieha za pomoci ICMPv6 správ nazývaných RS (Router Solicitation) a RA (Router Advertisement). Správa RS je zasielaná len na adresu FF02::2 a je určená len pre smerovače. Na adresu FF02::1 zasielajú smerovače správu RA určenú pre koncové stanice a neposielajú túto správu mimo danú sieť. Smerovače vedia poskytnúť adresu predvolenej brány, keďže v rámci lokálnej sieti sami figurujú ako brána. Túto adresu koncová stanica prijme, pričom klientskú časť adresy si koncová stanica odvodzuje sama.

Pre automatické vygenerovanie klientskej časti adresy sa využíva spomínaná technika EUI–64. Princíp spočíva vo vygenerovaní unikátneho identifikátora rozhrania z MAC adresy daného rozhrania. MAC adresa je 48 bitový identifikačný údaj, ktorý ale musí byť rozšírený o ďalších 16 bitov. Tým pádom dostávame 64 bitové ID (Identification) rozhrania. Princíp tejto techniky je rozdelenie MAC adresy na dve polovice a medzi tento priestor vložiť hexadecimálnu hodnotu FFFE. Pre lepšiu názornosť tento proces ilustruje obrázok 4.3.

Každá koncová stanica pripojená do internetovej siete je odkázaná na služby DNS. Bezstavový DHCPv6 poskytuje klientom DNS adresy bez toho, aby si udržoval akýkoľvek stav o poskytnutí týchto informácií. Zariadenia, ktoré využívajú bezstavovú automatickú konfiguráciu adresných parametrov, zároveň používajú bezstavový DHCPv6 na získanie adries DNS serverov.



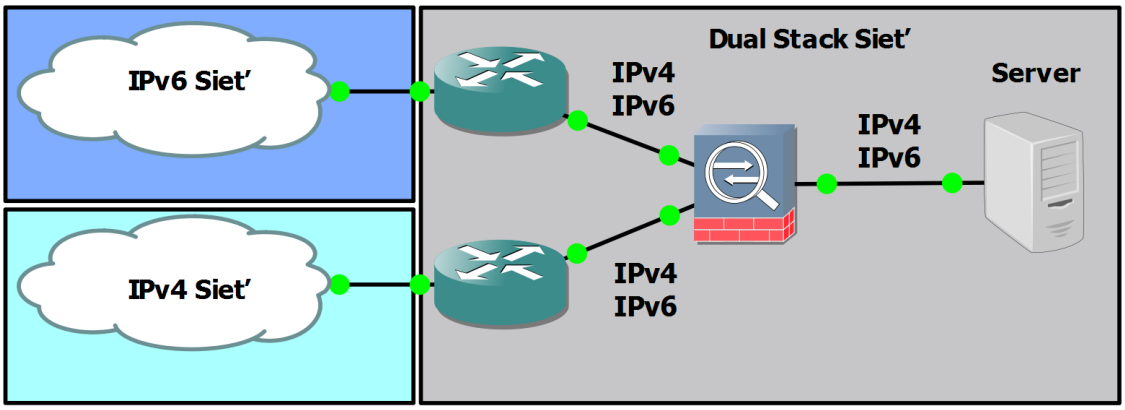
Obr. 4.3: Princíp procesu EUI–64 [30].

4.6 Tranzitné mechanizmy

Keďže protokol IPv4 je dlhé roky najpoužívanejším komunikačným protokolom, technicky nie je možné nasadiť nový IPv6 protokol globálne „zo dňa na deň“. Trvalé nasadenie novej protokolovej sady je potrebné riešiť sofistikovanými spôsobmi. Problematiku migrácie protokolov IPv4 a IPv6 môžeme definovať do skupín [25] [29]:

- Dual stack
- Tunelovanie
- Preklad protokolov

Dual stack – V prípade tejto techniky sa jedná o implementáciu oboch spomínaných protokolov v rámci infraštruktúry. Z toho vyplýva, že protokolové sady IPv4 a IPv6 pracujú súčasne a zdieľajú spoločne sieťové zdroje. Výhoda tejto metódy spočíva v jej jednoduchosti. Stačí nakonfigurovať zariadenia s podporou oboch protokolov. Určitou nevýhodou pri použití metódy dual stack je to, že všetky zariadenia musia podporovať protokolovú sadu IPv6. Obrázok 4.4 ilustruje techniku dual stack.



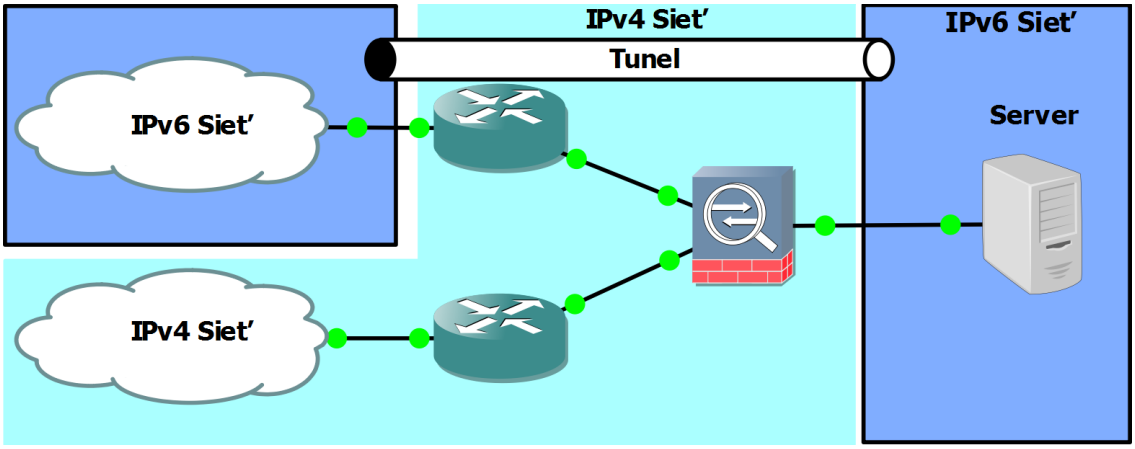
Obr. 4.4: Tranzitná technika – Dual stack [29].

Tunelovanie – Princípom tunelovacích techník je vytvorenie prechodnej siete (tunela), v ktorom sú IPv6 pakety zapuzdrené do IPv4 paketov alebo naopak. V prvom prípade sa potom tento zapúzdrený paket prenáša klasickým spôsobom cez IPv4 sieť. Takýmto spôsobom je možné využívať novú protokolovú sadu IPv6 v rámci starej IPv4 bez toho, aby sa táto oblasť musela nejako špeciálne modifikovať a prispôsobovať novému štandardu. Výhodou tejto techniky je minimálna konfigurácia zariadení, s čoho vyplýva menšie riziko chybovosti. Nevýhoda tunelovacích techník spočíva práve v tom, že oblasť IPv6 nevie priamo komunikovať s oblasťou IPv4. Na obrázku 4.5 je ilustrovaná technika tunelovania a tabuľka 4.3 zobrazuje rozdelenie tunelovacích protokolov [13] [30].

MCT (Manually Configured Tunnels) a GRE (Generic Routing Encapsulation) techniky sú navzájom veľmi podobné. Zaraďujú sa medzi statické IPv6 tunely typu

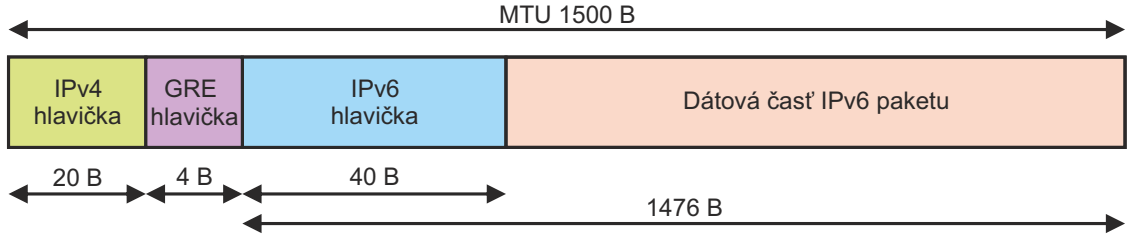
Tab. 4.3: Rozdelenie tranzitných tunelov.

Dynamické	ISATAP (Intra–Site Automatic Tunnel Addressing Protocol) 6to4 TEREDO
Statické	MCT GRE



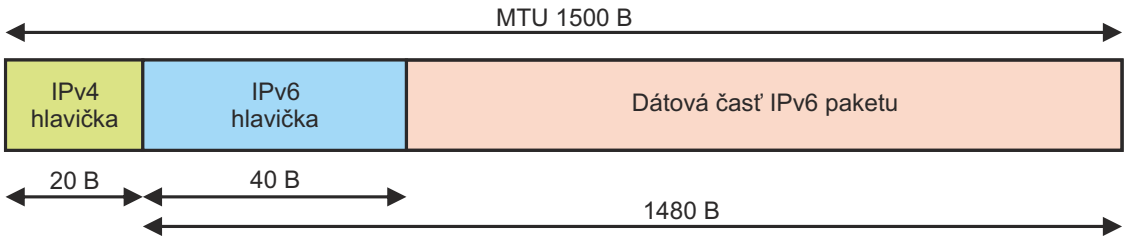
Obr. 4.5: Tranzitná technika – Tunelovanie [29].

bod–bod [30]. Treba však podotknúť, že zapúzdrovanie GRE nie je striktne tranzitná tunelovacia technika. Táto technika sa dá použiť na zapúzdrenie a prenos rôznych iných protokolov. Základnou črtou statických tunelov je konfigurácia zdroja a cieľa tunelu. Tieto parametre sú definované buď konkrétnou IPv4 adresou alebo samotným fyzickým rozhraním na daných smerovačoch. V prípade konkrétnej IP adresy sa najčastejšie využíva adresa loopback rozhrania. GRE tunelovanie je charakteristické hlavičkou, ktorá sa vkladá medzi hlavičky IPv4 a IPv6 paketov. Táto 4 bajtová hlavička definuje prenášaný protokol. Konečná veľkosť MTU zapúzdreného paketu je 1476 bajtov. Z toho 40 bajtov tvorí hlavička IPv6 paketu. Obrázok 4.6 ilustruje zapúzdrenie pomocou techniky GRE.



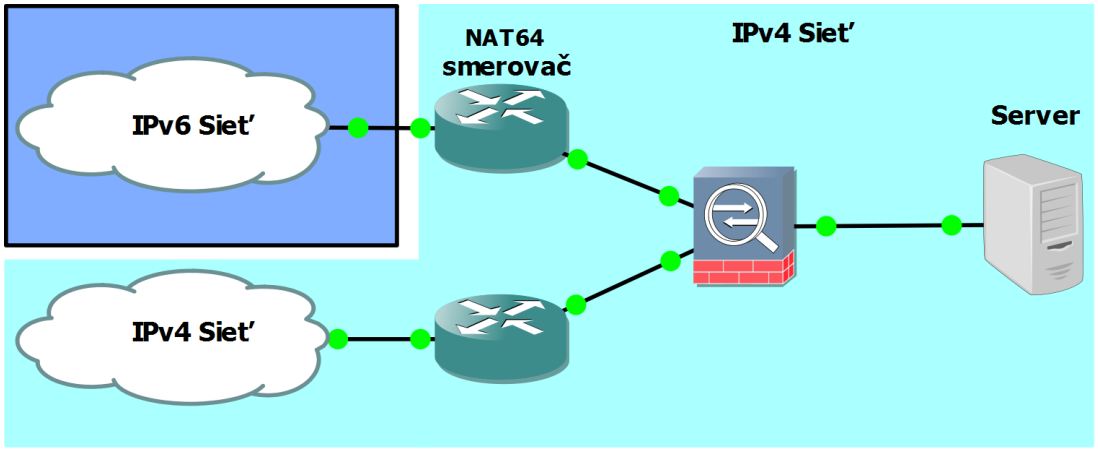
Obr. 4.6: Zapúzdrenia pomocou GRE [13].

ISATAP a 6to4 tunely sa radia medzi dynamické tunely typu bod–multibod [30]. V rámci týchto dynamických tunelov sa nastavuje iba zdrojová adresa tunelu. Pokiaľ budeme hovoriť o zapúzdrovaní 6to4, tak IPv6 adresu rozhrania tunelu bude tvoriť 48 bitový prefix. Prvý kvartál prefixu bude reprezentovať hodnota 2002, ktorá je typická pre adresy 6to4 tunelov. Druhý a tretí kvartál prefixu tvorí hexadecimálna hodnota IPv4 adresy zdroja tunelu. Aj v tomto prípade sa môže jednať o IPv4 adresu fyzického rozhrania alebo rozhrania loopback. Zvyšnú časť adresy tvoria nuly. Obrázok 4.7 ilustruje zapúzdrenie pomocou techník ISATAP a 6to4.



Obr. 4.7: Zapúzdrenia pomocou ISATAP / 6to4 [13].

Preklad protokolov – Táto technika umožňuje priamu komunikáciu medzi IPv6 a IPv4 koncovými zariadeniami [29]. Princípom mechanizmu je preklad IP adries komunikujúcich staníc v rámci IPv4 a IPv6 protokolu. Výhodou tejto techniky je plynulé nasadenie novej protokolovej sady do existujúcej infraštruktúry s minimálnou modifikáciou zariadení. Obrázok 4.8 ilustruje techniku prekladu.



Obr. 4.8: Tranzitná technika – Preklad protokolov [29].

4.7 Bezpečnosť v IPv6

Problematika bezpečnosti protokolu IPv6 je veľmi rozsiahla a preto ju nie je možné v tejto práci detailne rozobrať. Kapitola sa stručne zameriava na protokolovú sadu IPsec a v jednoduchosti popisuje princíp a využitie Cisco IOS firewallu.

S pomocou protokolu IPsec dokážeme vytvoriť zabezpečený tunel medzi dvomi sieťami resp. medzi dvomi stanicami, ktoré sú oddelené nedôveryhodnou sieťou [13]. Dokáže zabezpečiť autentizáciu a utajenie správ komunikujúcich strán. Implementácia IPsec v rámci IPv6 je povinná, zatiaľ čo v rámci IPv4 je to voliteľná súčasť, ktorá si našla uplatnenie. Ako bolo avizované v predošlých častiach, IPsec je sada protokolov zabezpečujúcich bezpečnú komunikáciu. Jedny z najvyužívanejších protokolov tejto sady sú:

- AHP (Authentication Header Protocol)
- ESP (Encapsulating Security Payload)

Medzi základné vlastnosti protokolu AHP patria, zaistenie overenia integrity prenášaných dát a overenia autenticity odosielateľa [13]. Protokol ESP slúži na šifrovanie dátovej časti paketov. Protokol IPsec sa používa v dvoch režimoch. Jeden z nich je tunelový režim, ktorého charakteristika spočíva v zapuzdrení dát do nového IP paketu. Druhá varianta sa nazýva transportná a jej význam spočíva v zachovaní pôvodnej hlavičky paketu a upravuje sa len dátová časť prenášaných paketov.

Protokoly ISAKMP (Internet Security Association Key Management) a IKE (Internet Key Exchange) slúžia na bezpečnú výmenu šifrovacích kľúčov [13]. Proces protokolu ISAKMP sa delí na dve fázy. V prvej fáze sa zabezpečí to, aby proces výmeny kľúčov bol chránený. Bezpečnosť v prvej fáze je riešená pomocou RSA (Rivest–Shamir–Adleman) algoritmu. V druhej fáze si komunikujúce strany vytvoria tajný symetrický kľúč za pomoci DH (Diffie–Hellman) algortimu.

Cisco IOS firewall je funkcia stavového firewallu, ktorá je poskytovaná v Cisco zariadeniach v rámci balíčku *Advanced Security* a vyššie. Jedným z možných využití spočíva vo vytvorení prístupového zoznamu, ktorý bude zahadzovať všetku prevádzku, ktorá pochádza z vonkajšej siete [13]. Výnimku bude tvoriť len prevádzka, ktorá bola iniciovaná z vnútornej siete. Zároveň táto prevádzka musí byť definovaná v stavovom firewalle, na základe čoho sa bude vytvárať relácia, podľa ktorej zariadenie vie rozpoznať prevádzku, ktorá pochádza pôvodne z vnútornej siete. Avšak pri protokole IPv6 treba brať do úvahy fakt, že nie je vhodné blokovať určité ICMPv6 správy, práve kvôli ich dôležitosť. Aj preto je implicitné ukončenie IPv6 prístupových zoznamov realizované položkami, ktoré povoľujú správy protokolu NDP. Až po týchto záznamoch sa nachádza položka, ktorá implicitne zakazuje celú prevádzku. Navyše Cisco IOS firewall ponúka možnosti ochrany siete proti škodlivému kódu a iným bezpečnostným rizikám [3].

5 SMEROVACIE PROTOKOLY

Kapitola stručne popisuje IGP smerovacie protokoly OSPFv2 (Open Shortest Path First version 2) a EIGRP (Enhanced Interior Gateway Routing Protocol). Zároveň sú uvedené kľúčové rozdielnosti medzi ich ekvivalentami OSPFv3 (Open Shortest Path First verion 3) a EIGRPv6 (Enhanced Interior Gateway Routing Protocol version 6) v rámci IPv6.

5.1 Protokol OSPFv2

Protokol OSPFv2 je klasifikovaný ako link–state protokol. Protokoly tohto typu majú obraz o celej topológii siete [25] [30]. Tento obraz sa nazýva link–state databáza, z ktorej sa pomocou SPF (Shortest Path First) algoritmu vypočíta tzv. SPF strom. Z tohto stromu sa následne vyberajú najlepšie cesty do smerovacej tabuľky. Aktualizácie sú rozosielané periodicky alebo v dôsledku zmeny v sieti.

Výhodou OSPF je možnosť rozdelenia infraštruktúry na niekoľko oblastí. Tieto oblasti môžeme rozdeliť do skupín:

- Štandardná oblasť
- Oblasť jadra
- Stub oblasť
- Totally stubby oblasť
- Not–so–stubby oblasť

Oblasť jadra je označovaná ako *Area 0*. Všetky ostatne oblasti musia mať priamu konektivitu s jadrom. Pokiaľ to nie je možné, riešenie ponúka virtuálna linka, ktorá tieto oblasti bude spájať.

Jednotlivé smerovače, na ktorých je aplikovaný OSPF komunikujú v rámci protokolu na základe správ, ktoré sa zasielajú na adresu 224.0.0.5:

- Hello
- Database Description
- Link–State Request
- Link–State Update
- Link–State Acknowledgment

Medzi susednými smerovačmi musí byť nadviazané susedstvo správami typu hello. Smerovač, ktorý obdrží hello správu, si zaznamená informácie a pošle správu ďalej. Na všesmerovej sieti ako je napríklad Ethernet sa komunikuje v rámci DR (Designated) a BDR (Backup Designated Router) smerovačov na adrese 224.0.0.6.

Autentifikáciu jednotlivých správ je možné riešiť pomocou nešifrovaného reťazca znakov alebo šifrovaného reťazca znakov pomocou hešovacieho algoritmu MD5 (Message–Digest Algorithm 5). Údaje o šifrovaní obsahuje hlavička paketu OSPF.

5.2 Protokol EIGRP

Protokol EIGRP je označovaný aj ako hybridný protokol, pretože disponuje určitými vlastnosťami link–state a distance–vector protokolov [25] [30]. Protokol je proprietárny a je použiteľný len na Cisco zariadeniach. Smerovač s nakonfigurovaným EIGRP má obraz o celej sieti a do smerovacej tabuľky sú vložené najlepšie cesty, ktoré sú vypočítané na základe DUAL (Diffusing Update Algorithm) algoritmu. Smerovací protokol zasiela aktualizácie len pri určitej zmene v sieti. Protokol je vhodný aj pre komplexnejšie siete.

EIGRP nedisponuje vlastnosťou rozdelenia sieťovej domény na menšie celky, ako to bolo u OSPF pomocou rôznych typov oblastí.

Pokiaľ sa jedná o nadviazanie susedstva, protokol EIGRP sa tak isto opiera o hello pakety. Tento proces pracuje obdobne s tým rozdielom, že hello pakety sa zasielajú na skupinovú adresu 224.0.0.10.

V rámci autentizácie správ EIGRP podporuje iba hešovací algoritmus MD5. Nedisponuje s autentizáciou na základe nešifrovaného textu.

5.3 Smerovacie protokoly IPv6

Podstata a princíp smerovacích protokolov ostal rovnaký, avšak IPv6 prináša isté rozdiely. Výber niektorých zmien je obsiahnutý v nasledujúcich riadkoch. Čerpané bolo zo zdrojov [8], [23] a [30].

Smerovací protokol OSPFv3

- Pridanie sietí v konfiguračnom režime rozhrania. Absencia príkazu **network**.
- Susedstvá sú nadviazané v rámci linkovej lokálnej adresy.
- Identifikácia susedstva na základe *Router ID*.
- Nové LSA (Link–State Advertisement) správy. (Link LSA, Intra–area Prefix LSA)
- Možnosť prevádzkovať niekoľko OSPFv3 inštancií v rámci jedného rozhrania.
- Autentizácia správ na základe IPsec.
- Zmena niektorých polí OSPF hlavičky.
- Využíva skupinové adresy FF02::5 a FF02::6.

Smerovací protokol EIGRPv6

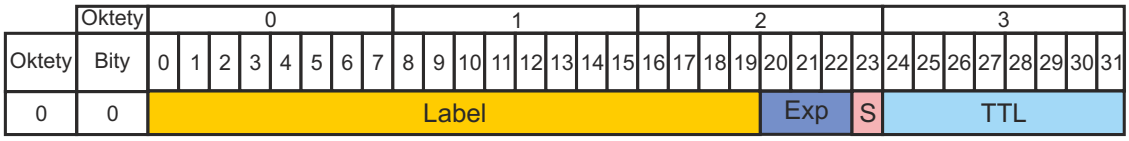
- Pridanie sietí v konfiguračnom režime rozhrania. Absencia príkazu **network**.
- Nadviazanie susedstva je možné aj pokiaľ smerovače nefigurujú v jednej sieti.
- Príkaz **(no)shutdown** na spustenie a zastavenie EIGRP procesu.
- Filtrovanie ciest sa realizuje pomocou príkazu **distribute–list** a **prefix–list**. Príkaz **route–map** nie je podporovaný.
- Využíva skupinovú adresu FF02::A.

6 PROTOKOL MPLS

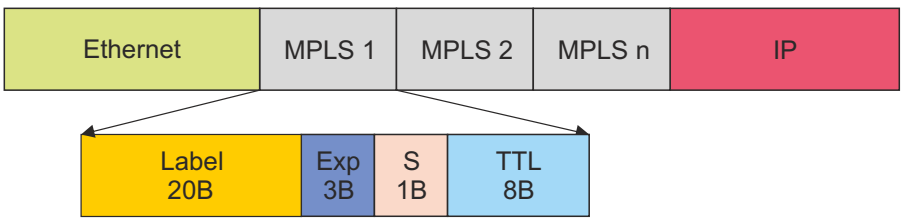
Kapitola stručne pojednáva o technológií MPLS. Úvod kapitoly v krátkosti približuje túto technológiu. Ďalej je v kapitole zahrnutý princíp tejto technológie a záver obsahuje problematiku MPLS v IPv6.

6.1 Úvod do MPLS

MPLS je prepínací mechanizmus, ktorý k preposielaniu dát používa takzvané štítky (labels) [28]. Obrázok 6.1 ilustruje hlavičku protokolu MPLS. Pole *Label* reprezentuje číselnú hodnotu štítku, resp. cieľovú adresu. Na sieťovej vrstve k tomuto účelu slúži IP adresa. Pole *Exp* slúži na experimentálne účely. Momentálne je toto pole využívané ako CoS (Class of Service). Keďže medzi druhou a treťou vrstvou môžeme mať niekoľko štítkov, je definované pole *S*, ktoré identifikuje posledný štítok. Posledné pole je *TTL*, ktoré reprezentuje dobu života štítku. MPLS bolo navrhnuté tak, aby bolo nezávislé na použitej sieťovej technológií. To znamená, že MPLS dokáže pracovať ako s klasickým Ethernetom tak aj s Frame Relay alebo ATM. V rámci zapúzdrenia sa štítok nachádza medzi druhou a treťou vrstvou sieťového modelu ISO/OSI ako ilustruje obrázok 6.2. Aj práve z tohto dôvodu sa MPLS označuje ako protokol vrstvy 2,5 (Layer 2.5). Prepínanie paketov je založené na informáciach, ktoré sú získané zo sieťovej a vyšších vrstiev. Toto umožňuje zrýchliť samotnú komunikáciu oproti bežnému IP smerovaniu, pretože zariadenia sa nemusia zaoberať vyššími vrstvami. Takto odpadáva napríklad vyhľadávanie cieľovej cesty v smerovacej tabuľke.



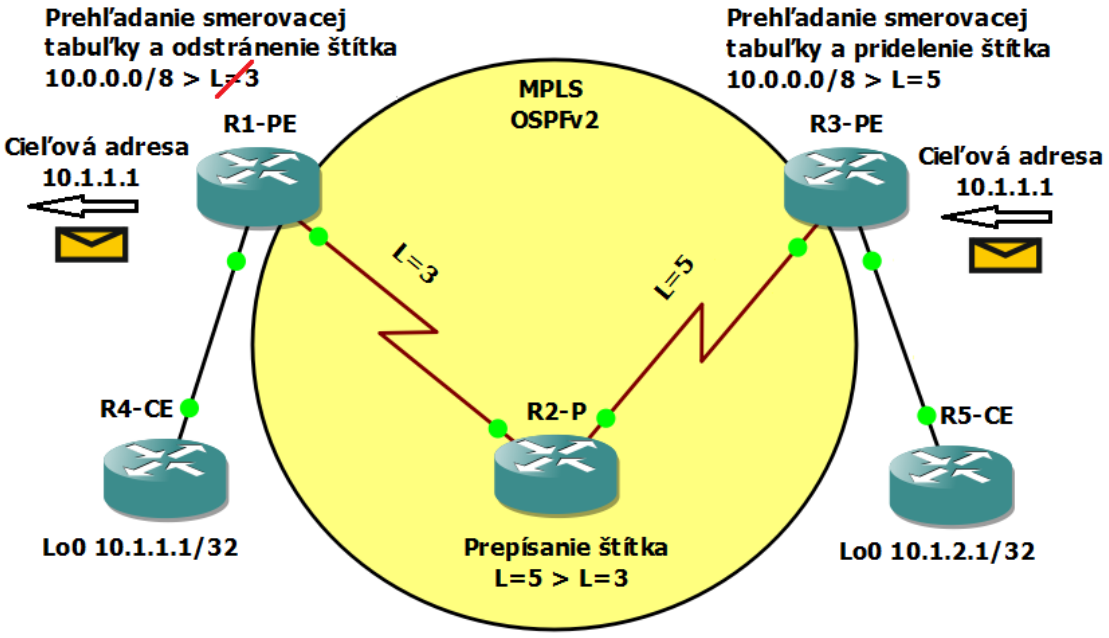
Obr. 6.1: Hlavička protokolu MPLS [28].



Obr. 6.2: Zapúzdrovanie hlavičiek MPLS [28].

6.2 Princíp MPLS

MPLS oblasť sa skladá zo siete smerovačov, ktorá obsahuje dve základne entity a to PE (Provider Edge) a P (Provider) [28]. Tieto smerovače sa môžu nazývať aj ELSR (Edge Label Switch Router) a LSR (Label Switch Router). CE (Customer Edge) nie je priamou súčasťou MPLS siete. PE resp. ELSR vykonáva prehľadávanie smerovacej tabuľky a pridelenie štítku. Smerovače typu P resp. LSR vykonávajú prepínanie hodnôt štítkov. Toto je ilustrované obrázkom 6.3. Smerovač R3 prijal správu, ktorá má byť zaslaná na adresu 10.1.1.1. Smerovač vykoná bežné vyhľadanie cesty v smerovacej tabuľke a následne pridelý štítok s číselnou hodnotou L = 5. Smerovač R2 na základe štítku s hodnotou L = 5 zasiela správu ďalej smerovaču R1 a zároveň prepíše hodnotu štítku na hodnotu L = 3. Smerovač R1 prijme túto správu, odstráni štítok a zasiela správu na základe nájdenej cesty zo smerovacej tabuľky.



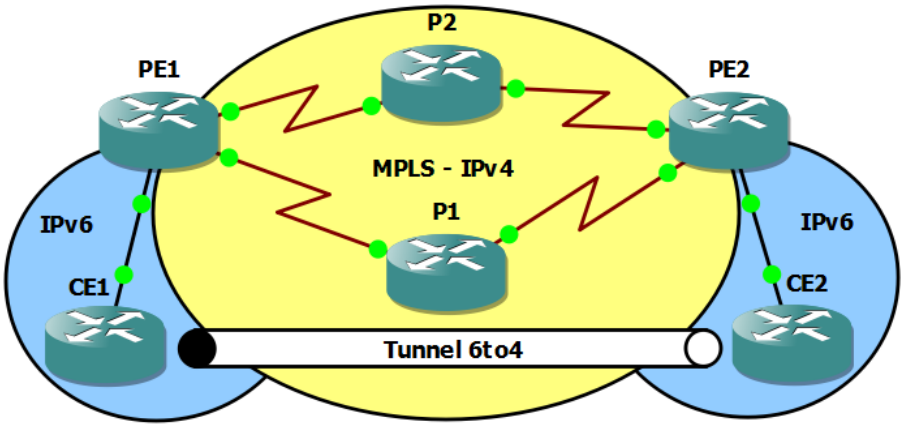
Obr. 6.3: Princíp technológie MPLS [28].

MPLS sa skladá z dvoch hlavných komponentov a tými sú riadiaca a dátová rovina [28]. Riadiaca rovina zahŕňa v sebe mechanizmi na výmenu smerovacích údajov. Na toto slúžia práve smerovacie protokoly ako OSPF, EIGRP, BGP (Border Gateway Protocol) a iné. Súčasťou riadiacej roviny je aj výmena informácií o štítkoch. Na to slúžia protokoly ako TDP (Tag Distribution Protocol) a LDP (Label Distribution Protocol). Dátová rovina slúži na samotné zasielanie správ či už na základe cieľovej IP adresy alebo štítku. Tabuľka LFIB (Label Forwarding Information Base) slúži na posielanie správ na základe informácií obsiahnutých v štítku. LFIB tabuľka sa napĺňa údajmi na základe protokolov použitých v riadiacej rovine.

6.3 MPLS a IPv6

Mnoho poskytovateľov, ktorý majú založené jadro siete na technike MPLS, má svoju infraštruktúru založenú na báze IPv4 protokolov [16]. MPLS je nefunkčné pokiaľ sú zariadenia nakonfigurované iba s využitím IPv6 protokolov. Dôvod je ten, že správy LDP protokolu nie sú distribuované IPv6 smerovacími protokolmi. V súčasnej dobe sa táto problematika rieši, o čom svedčí aj návrh *Updates to LDP for IPv6* [26].

V súčastnosti je niekoľko rôznych spôsobov, ktoré môžeme využiť k tomu aby sme tento problém čiastočne vyriešili. Jeden zo spôsobov je využiť tunelovaciu techniku typu 6to4 [16]. Princípom je vytvorenie tunelu medzi hraničnými smerovačmi zákazníkov. To zahŕňa iba konfiguráciu smerovačov typu CE na strane zákazníka. Toto riešenia sa nijako nedotkne MPLS smerovačov a tým pádom ani smerovačov na strane poskytovateľa. Obrázok 6.4 ilustruje princíp riešenia tejto techniky.



Obr. 6.4: Princíp MPLS v IPv6 – tunel 6to4.

Ďalšou z možností je využiť MP–BGP (Multiprotocol BGP) protokol [16]. Týmto spôsobom sa konfigurácia dotkne len hraničných smerovačov poskytovateľa. Na smerovačoch typu PE sa nakonfiguruje iBGP (Internal BGP) susedstvo. Povolí sa funkcia umožňujúca BGP smerovačom spracovávanie IPv6 protokolu. Manuálne sa nadviaže spojenie medzi BGP smerovačmi a povolí sa zasielanie MPLS štítkov. Výmenu IPv6 smerovacích informácií umožní MP–BGP, pričom IPv6 pakety sa zasielajú MPLS štruktúrou pomocou dvoch štítkov. Jeden štítok sa použije na dosiahnutie hraničného smerovača PE. Druhý štítok slúži na identifikáciu IPv6 paketu. Toto zapúzdrenie je veľmi jednoduchou formou ilustrované obrázkom 6.5.

Obr. 6.5: Princíp MPLS v IPv6 – technika 6PE [11].

7 PRAKTICKÁ REALIZÁCIA ZADANIA

Kapitola je rozdelená na tri časti. Prvá časť pojednáva o simulačnom prostredí GNS3 a obsahuje základný popis konfigurácie prostredia. Druhá časť popisuje prácu v prostredí GNS3. Tretia časť tejto kapitoly obsahuje popis laboratórnych úloh. Samotné návody k laboratórnym úlohám sú k dispozícií v prílohe diplomovej práce. Konfiguračné súboru jednotlivých zariadení sa nachádzajú na priloženom DVD.

7.1 Nastavenie prostredia GNS3

Laboratórne úlohy sú koncipované pre verziu GNS3 v1.3.1 a platformu operačného systému Windows 7. V nasledujúcich častiach sú ukážky a popis základných počiatočných nastavení prostredia GNS3.

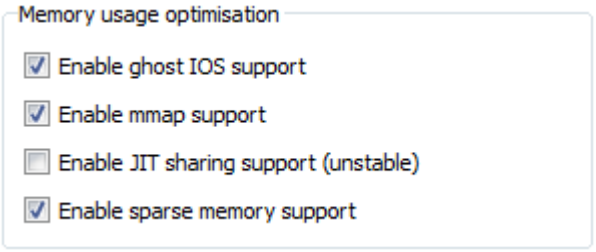
7.1.1 Základné nastavenia

V hornej lište z ponuky *Edit* otvoríme sekciu *Preferences*. V záložke *General* si môžeme skontrolovať kam sa ukladajú projekty, dočasné súbory a kde sú uložené IOS obrazy. Implicitné nastavenia sú znázornené na obrázku 7.1, kde údaj XY je názov používateľa operačného systému Windows 7.

Obr. 7.1: Záložka General.

7.1.2 Nastavenie Dynamips

V hornej lište z ponuky *Edit* otvoríme sekciu *Preferences*. Záložka *Dynamips* obsahuje rozšírené nastavenia emulátoru Dynamips. V záložke *Advanced settings* sa nachádzajú funkcie ako *ghost IOS*, *mmap* a *sparse memory*. Tieto funkcie boli popísané v kapitole 1.3 na strane 18. Ukážka je ilustrovaná na obrázku 7.2.



Obr. 7.2: Záložka Dynamips – Advanced settings.

7.1.3 Importovanie IOS súborov

V kapitole 7.1.1 na strane 49 sa nachádza popis cesty do zložky s IOS obrazmi. V tejto zložke by sa mali nachádzať operačné systémy k smerovačom, ktoré hodláme používať. V tomto bode je vhodné upozorniť na to, že IOS obrazy môžeme umiestniť do tejto zložky ako nerozbalené súbory s príponou *bin* alebo si ich rozbalíme a budeme používať rozbalené súbory s príponou *image*. Je vhodné zvoliť druhú variantu. Pokiaľ by sme súbory nerozbalili, po každom spustení smerovača by dochádzalo k rozbaľovaniu IOS obrazu. Tento proces je samozrejme zbytočný a časovo náročný.

Pokiaľ máme zdrojové súbory v danej zložke, môžeme pristúpiť k samotnému pridruženiu špecifického IOS obrazu ku konkrétnemu zariadeniu. V hornej lište z ponuky *Edit* otvoríme sekciu *Preferences*. Rozklikneme sekciu *Dynamips*. V novo rozbalenej záložke *IOS routers* klikneme na položku *New*. Otvorí sa inštalačná ponuka, v ktorej je nutné zadať cestu ku konkrétnemu IOS obrazu. V ďalších ponukách inštalácie môžeme osadiť smerovač rôznymi rozhraniami v závislosti na použitej rade smerovača. Kompatibilita konkrétnych modulov sa nachádza na internetových stránkach programu GNS3 [12]. Ďalej môžeme vybrať veľkosť pamäte RAM pre daný smerovač. Každopádne GNS3 má implicitne nakonfigurované minimálne hardwarové požiadavky pre dané zariadenia. Ďalšou možnosťou v rámci inštalácie sa ponúka vygenerovanie *idlepc* hodnoty. Dôležitosť tejto hodnoty bola popísaná v kapitole 1.3 na strane 18.

7.1.4 Koncové zariadenia v GNS3

Pokiaľ potrebujeme implementovať do nami vytvorenej siete koncové zariadenia, môžeme to zrealizovať niekoľkými spôsobmi [1].

- Smerovač
- VPCS (Virtual PC Simulator)
- Loopback rozhranie
- VMware / VirtualBox klient

Smerovač – Jednou z možností je pridať ďalší smerovač, ktorý bude figurovať ako klasická koncová stanice. To znamená, že na ňom musíme vykonať isté minimálne konfiguračné zmeny ako sú uvedené na nasledujúci riadkoch [1].

```
Router(config)#no ip routing                          !vypne funkciu smerovania
Router(config)#interface fa0/0                        !konfigurácia rozhrania
Router(config-if)#ip address adresa maska             !adresa a maska siete
Router(config-if)#no shutdown                         !zapnutie rozhrania
Router(config-if)#exit          !návrat do globálneho konfiguračného režimu
Router(config)#ip default-gateway adresa              !adresa predvolenej brány
```

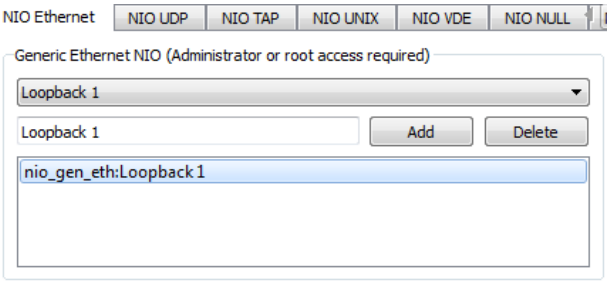
Nevýhodou tejto varianty je pridanie redundantného smerovača, ktorý má za následok väčšie vyťaženie výpočetných zdrojov. Práve preto sa táto varianta doporučuje s využitím smerovača najnižšej rady, ktorú GNS3 podporuje a tým je smerovač Cisco 1700. Aj napriek tomu sa ale tento spôsob doporučuje len ako posledná možnosť.

Virtuálny PC simulátor – Táto funkcia je prístupná z ponuky koncových staníc, ktorú nájdeme na ľavej časti pracovného prostredia GNS3. VPCS môžeme využiť hlavne v prípadoch, kde sa nám jedná o jednoduché testovanie siete s príkazmi **ping**, **trace** alebo **rlogin**. V prípade príkazu **trace**, hovoríme o štandardnom príkaze **tracert**, ktorý používá systém Windows. Príkaz **rlogin** slúži na vzdialenú správu sieťových zariadení. Ekvivalentom tohoto príkazu v systéme Windows je príkaz **telnet**. Po spustení VPCS klienta si následne môžeme spustiť konzolu, ktorá nám otvorí VPCS prostredie. Na výber máme niekoľko možností konfigurácie. Práca v konzolovom okne je intuitívna a prostredie nám tiež ponúka nápovedu. Určitou nevýhodou VPCS je limitovaný počet koncových staníc, ktoré je schopné simulovať. VPCS totiž pracuje maximálne s deviatimi stanicami. Ďalšou nevýhodou je absencia možnosti fragmentovaného prenosu dát a slabá podpora protokolu IPv6.

Loopback rozhranie – Ďalšou možnosťou, ktorá je viac sofistikovaná je vytvoriť MS (Microsoft) loopback sieťový adaptér. Používateľ má niekoľko možnosti ako to zrealizovať. Jednou z nich je pridať toto rozhranie v prostredí operačného systému Windows. Toto je možné v ovládacom panely zo sekcie *Správca zariadení*. Po nainštalovaní nového rozhrania je nutný reštart počítača. Pokiaľ chceme využívať loopback rozhranie v prostredí GNS3, je nutné spúšťať program ako správca (*Run as Administrator*).

Z ponuky koncových staníc na ľavej časti okna vyberieme a presunieme do pracovného prostredia objekt *cloud*. Pravým tlačidlom vyvoláme kontextové menu, z ktorého vyberieme *Configure*. V záložke *NIO Ethernet* z ponuky *Generic Ethernet NIO* vyberieme loopback rozhranie. Toto ilustruje obrázok 7.3. Klikneme na tlačidlo

Add a následne *Apply*. Po tomto kroku môžeme pripojiť loopback rozhranie, ktoré je reprezentované objektom *cloud* do simulovanej siete. Adresáciu loopback rozhrania je možné nakonfigurovať automaticky pomocou DHCP protokolu pokiaľ sa v simulovanej sieti nachádza DHCP server. Druhou možnosťou je adresy nakonfigurovať manuálne v operačnom systéme v rámci nastavenia adaptéru. Takto sa hostiteľský počítač stane súčasťou virtualizovaného prostredia.

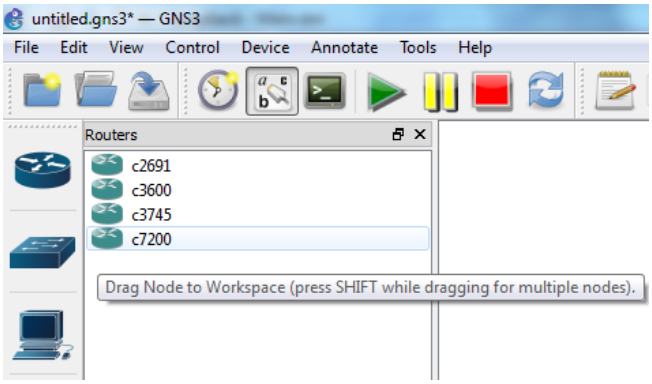


Obr. 7.3: Nastavenie objektu cloud.

VMware / VirtualBox klient – Táto varianta je istým spôsobom vylepšenou obdobou možnosti loopback rozhrania popísaného vyššie. Pomocou týchto klientov sme schopní virtualizovať operačné systémy ako Ubuntu, Windows, JunOS alebo aj CheckPoint. V rámci týchto klientov sa vytvorí virtuálne sieťové rozhranie, s ktorým sa pracuje rovnako ako s predošle popísaným loopback rozhraním.

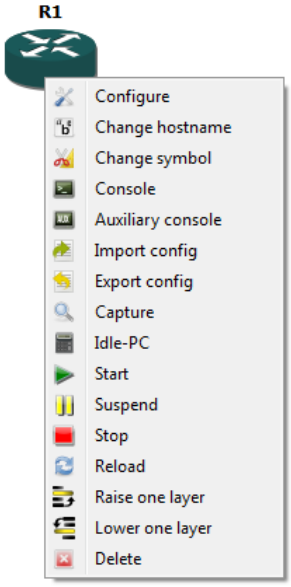
7.2 Práca v prostredí GNS3

Pre základnú prácu si vystačíme s pár jednoduchými pravidlami. Pokiaľ chceme pridať do pracovného prostredia smerovač, jednoducho ho presunieme z výberu na pravej časti palety. Rozklikneme si sekciu smerovačov a vyberieme nami žiadanú radu. Tento krok je ilustrovaný na obrázku 7.4



Obr. 7.4: Výber smerovača.

Pokiaľ máme smerovač na pracovnej ploche, klikneme pravým tlačidlom myši na zariadenie. Zobrazí sa nám ponuka, z ktorej máme na výber niekoľko užitočných odkazov pre prácu so smerovačom. Tento krok ilustruje obrázok 7.5.



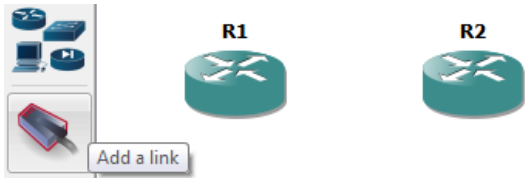
Obr. 7.5: Ponuka pre prácu so smerovačom.

Všetky tieto prvky sú obsiahnuté aj v hornej lište z ponuky *Device*. V tabuľke 7.1 je výber a popis niektorých prvkov tejto ponuky.

Tab. 7.1: Ponuka výberu operací na vybranom zariadení.

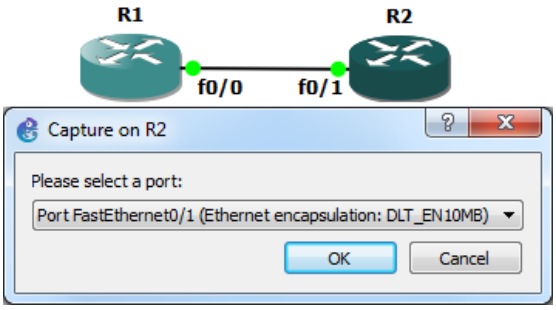
Configure	Umožňuje detailné nastavenia vlastnosti zariadenia. Tu sa nám ponúka konfigurácia veľkosti pamäti RAM, NVRAM (Non–Volatile Random Access Memory) a samotnej flash pamäte. Ďalej máme možnosť zvoliť zásuvné moduly v rámci kompatibility s daným zariadením a iné.
Console	Slúži na spustenie konzoly na zvolenom zariadení.
Capture	Slúži na spustenie zachytávania sieťovej prevádzky na konkrétnom rozhraní vybraného zariadenia. Na zachytávanie sa môže využiť napríklad program Wireshark.
Idle–PC	Slúži na vyhľadanie najlepšej *idlepc* hodnoty pre dané zariadenie resp. konkrétny IOS. Táto funkcia je vhodná pokiaľ procesor vykazuje vysoké vyťaženie a musíme vyhľadať lepšie hodnoty *idlepc*.
Start	Slúži na spustenie vybraného zariadenia.
Reload	Slúži na reštartovanie vybraného zariadenia.

Na prepojenie zariadení klikneme na ikonu konektora, ktorá sa nachádza v palete na ľavej strane. Toto ilustruje obrázok 7.6. Po kliknutí na konkrétne zariadenie sa zobrazí ponuka s dostupnými rozhraniami zariadenia. Po vybratí rozhrania klikneme na druhé zariadenie a vykonáme rovnaké kroky.

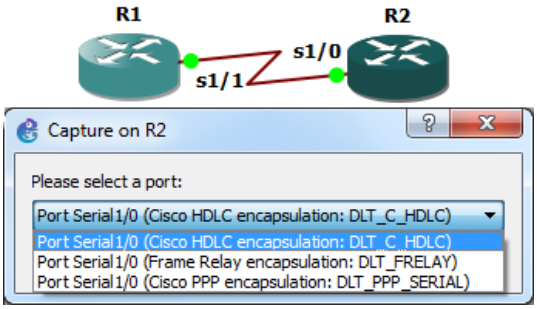


Obr. 7.6: Prepojenie dvoch zariadení.

Zachytávanie prevádzky je možné vďaka programu Wireshark, ktorý umožňuje zachytiť sieťovú prevádzku na jednotlivých portoch zariadení. Kliknutím pravým tlačidlom na zvolený smerovač vyvoláme kontextové menu a následne klikneme na funkciu *Capture*. Ako ilustruje obrázok 7.7, zvolíme konkrétny port smerovača R2, na ktorom chceme sledovať sieťovú prevádzku. Ako ilustruje obrázok 7.8, pri výbere sériového rozhrania máme na výber tri druhy zapúzdrovania a to Cisco PPP (Point–to–Point Protocol), Frame Relay a Cisco HDLC (High–Level Data Link Control). V laboratórnych úlohách sa používa implicitné zapúzdrovanie pre Cisco smerovače.



Obr. 7.7: Ponuka pre zachytávanie paketov na fastEthernetovom rozhraní.



Obr. 7.8: Ponuka pre zachytávanie paketov na sériovom rozhraní.

Zobrazenie popiskov rozhraní nájdeme pod hornou lištou. Na obrázku 7.9 je táto funkcia farebne vyznačená. Funkcia si nájde využitie hlavne pri rýchlej kontrole topológie v rámci prepojenia jednotlivých zariadení.

Obr. 7.9: Funkcia na zobrazenie popiskov rozhraní.

7.3 Laboratórne úlohy

Návody sú písané v českom jazyku kvôli lepšiemu porozumeniu pre českého študenta. V návodoch je použité mimo normálneho aj tučné písmo a kurzíva. Tučné písmo reprezentuje príkazy a kurzíva reprezentuje kľúčové slová. Kombinácia tučného písma a kurzívy predstavuje príkaz, ktorý obsahuje premenné parametre. Bližšie vysvetlenie použitia typov písma je zahrnuté v tabuľké 7.2.

Smerovače použité v laboratórnych úlohách sú Cisco 7200 a použitý IOS je Advanced Enterprise K9 verzia 15.2(4)M2. V úlohe číslo 2 sa nachádza smerovač Cisco 3600, ktorý figuruje ako prepínač. Toto je riešené tým, že je osadený EtherSwitch modulom. Tento smerovač používa IOS C3660–JK9O3S–M verzia 12.4(6)XT2. V druhej a tretej úlohe sa využívajú virtualizované stanice, na ktorých je nainštalovaný operačný systém Windows 7. Implementácia virtualizovaných staníc je riešená prostredníctvom programu VMware.

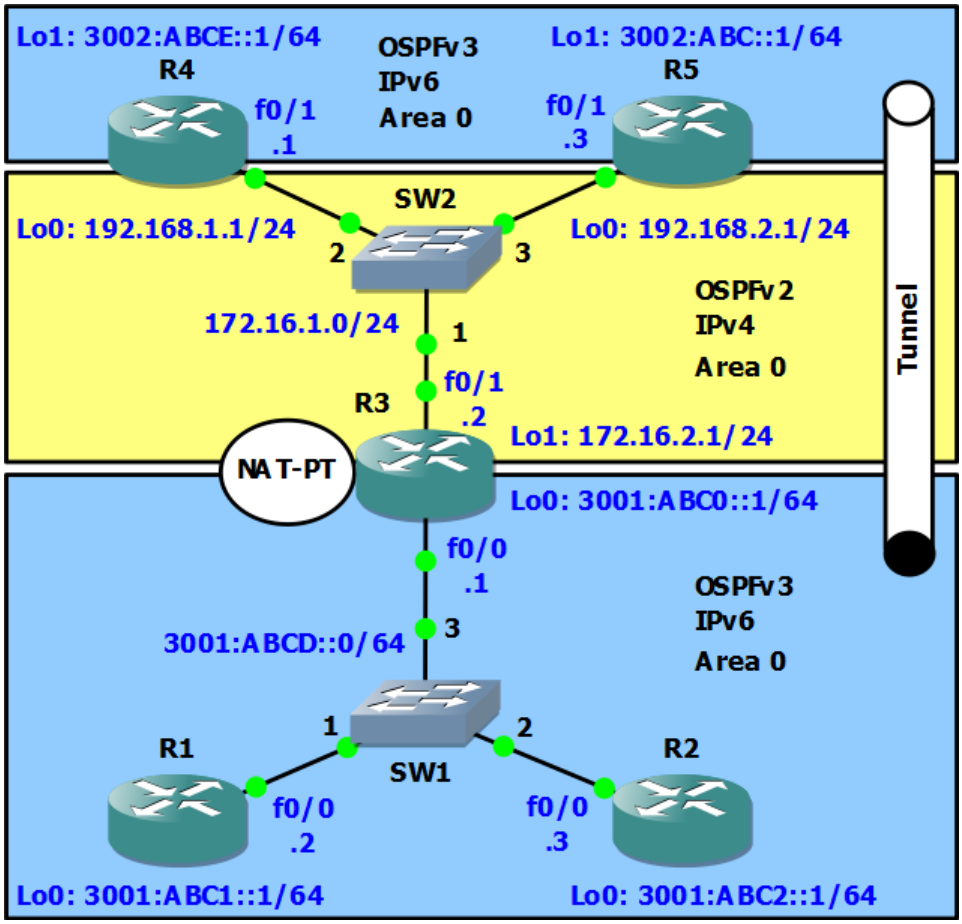
Pre lepšiu názornosť sa úlohy opierajú o rôzne výpisy IOS príkazov ako **show** a **debug**. Navyše je v úlohách zakomponovaná práca s programom Wireshark. Tým pádom sú v úlohách kvalitne znázornené rôzne procesy a princípy fungovania spomínaných protokolov a techník. Na konci každej laboratórnej úlohy sa nachádzajú kontrolné otázky. Na ich základe si študent môže overiť svoje vedomostné znalosti z danej problematiky. Navyše sa na konci každej laboratórnej úlohy nachádza samostatná doplnková úloha. Táto doplnková úloha vychádza z obsahu danej laboratórnej úlohy a študent by ju tak mal byť schopný vyriešiť.

Tab. 7.2: Príklady syntaxe v návodoch laboratórnych úloh.

show ip route	príkazy
ip address *1.1.1.1 255.255.255.255*	príkazy obsahujúce premenné parametre
Label Stack	kľúčové výrazy

7.3.1 Laboratórna úloha 1

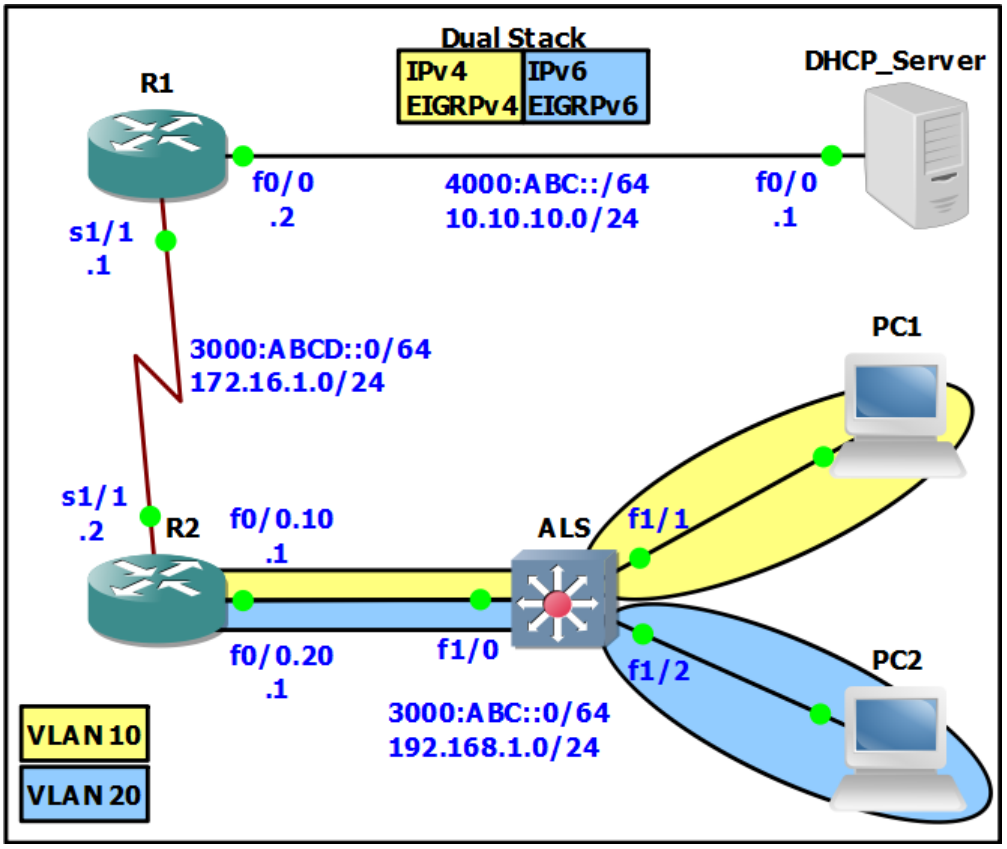
Obrázok 7.10 ilustruje zapojenie topológie siete. Laboratórna úloha sa v prvej časti zameriava na porovnanie protokolovej sady IPv4 a IPv6 v rámci smerovacích protokolov OSPFv2 a OSPFv3. Obsahuje aj náležitosti autentizácie správ OSPF. V rámci tejto časti úlohy je kladený dôraz hlavne na odlišnosti jednotlivých protokolov a ich praktickú konfiguráciu. Ďalej úloha zahŕňa problematiku tranzitných mechanizmov a to konkrétne techniky NAT–PT a tunelovanie. Aj napriek tomu, že NAT–PT je v súčastnosti považovaná za zastaralú techniku, v rámci edukačných účelov je vhodná na ukážku tejto tranzitnej techniky. V rámci tunelovacích techník sa v úlohe vyskytuje konfigurácia tunelu 6to4 a GRE. Návody k laboratórnej úlohe sú v prílohe A.1 na strane 72.



Obr. 7.10: Topológia zadania laboratórnej úlohy 1.

7.3.2 Laboratórna úloha 2

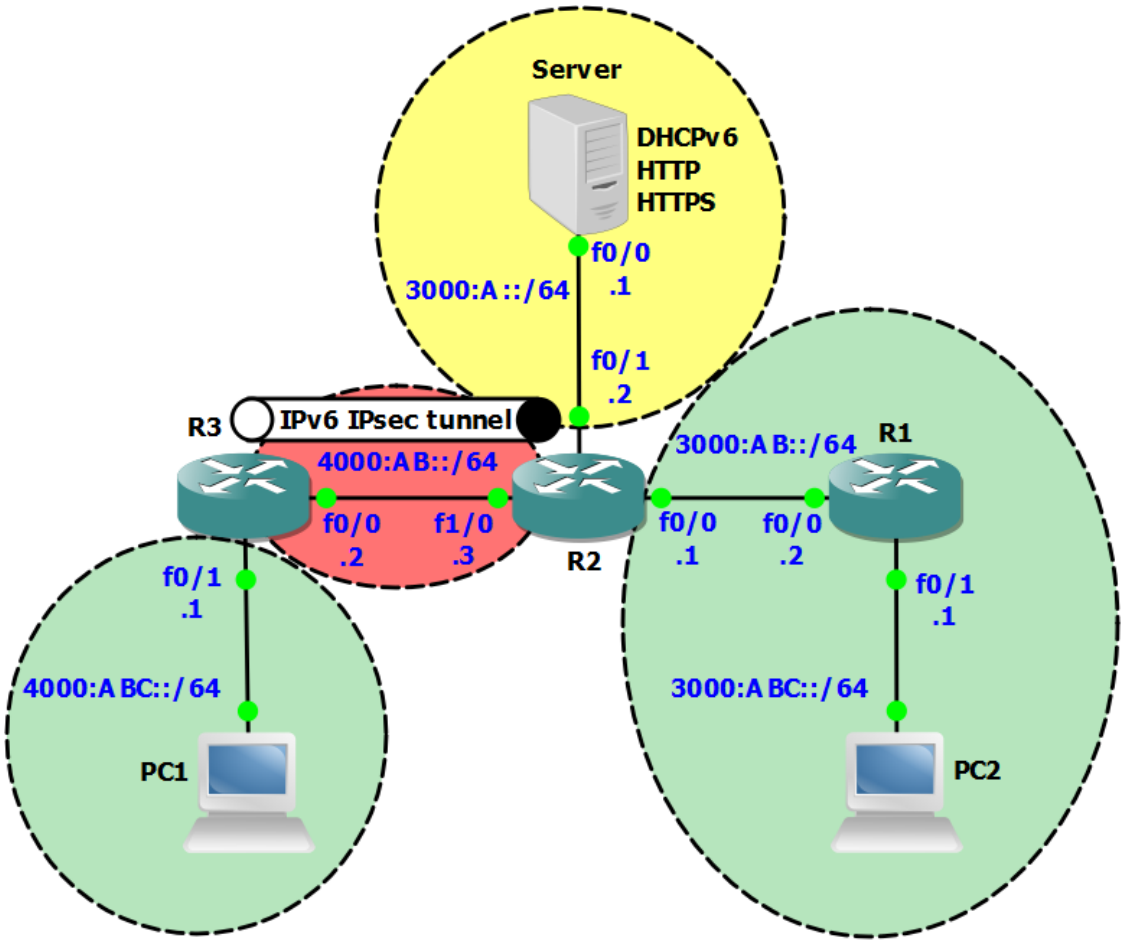
Obrázok 7.11 ilustruje zapojenie topológie siete. Prvá časť laboratórnej úlohy sa zameriava na konfiguráciu smerovacích protokolov EIGRP a EIGRPv6. Topológia je navrhnutá v zmysle dual stack. V tejto sieti sa nachádza smerovač, ktorý figuruje ako DHCP server. Ďalšou časťou úlohy je konfigurácia DHCPv4 a DHCPv6 služieb. Pre túto a ďalšie časti úlohy sa ako koncové stanice využívajú virtualizované stanice, na ktorých je nainštalovaný operačný systém Windows 7. Toto je riešené pomocou VMware klienta. VPCS nie je vhodný kvôli obmedzeným možnostiam. Jedným z nich je neúplná podpora DHCP funkcií v rámci IPv6 protokolu. Ďalším obmedzením je nemožnosť fragmentovaného prenosu dát. Posledná časť úlohy sa zaoberá problematikou protokolu ICMP. Táto časť úlohy pojednáva mimo iného aj o fragmentácií v sieti IPv4 a IPv6. Návody k laboratórnej úlohe sú v prílohe A.2 na strane 91.



Obr. 7.11: Topológia zadania laboratórnej úlohy 2.

7.3.3 Laboratórna úloha 3

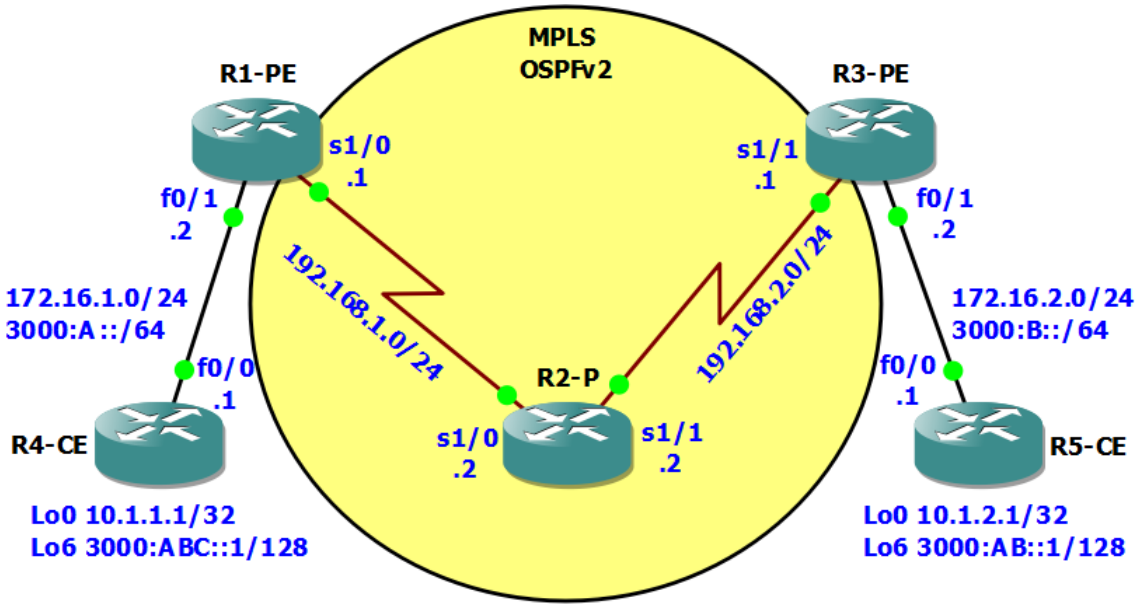
Obrázok 7.12 ilustruje zapojenie topológie siete. Účelom tejto úlohy je poukázať na bezpečnostné aspekty a riešenia v IPv6. Táto úloha poukazuje na niektoré zabezpečenia týkajúce sa sieťovej vrstvy, na ktorej pracuje samotný protokol IPv6. Sieť je rozdelená na niekoľko častí. Zelené oblasti sú považované za vierohodné siete. V žltej oblasti sa nachádza server, ktorý slúži ako DHCPv6, HTTP (Hypertext Transfer Protocol) a HTTPS (Hypertext Transfer Protocol Secure). Úloha v prvej časti pojednáva o konfigurácií smerovacieho protokolu OSPFv3. Ďalej sa v úlohe konfiguruje bezstavový DHCPv6 server. V úlohe sa tematika bezpečnosti delí na štyri časti. V prvej časti sa konfiguruje autentizácia OSPFv3 režijných správ. K tomuto sa využíva IPsec protokol. V ďalšej časti úlohy sa konfiguruje IPsec tunel. Tretia časť sa zameriava na konfiguráciu prístupových zoznamov. Kladie sa tu dôraz na filtrovanie správ protokolu ICMPv6. Posledná časť pojednáva o stavovom Cisco IOS firewalle. Návody k laboratórnej úlohe sú v prílohe A.3 na strane 108.



Obr. 7.12: Topológia zadania laboratórnej úlohy 3.

7.3.4 Laboratórna úloha 4

Obrázok 7.13 ilustruje zapojenie topológie siete. Prvá časť úlohy sa zameriava na základnú konfiguráciu techniky MPLS. V tejto časti úlohy sú prakticky vysvetlené princípy tejto techniky, dôležitosť parametru *Router ID* a autentizácia v rámci LDP spojenia. V tejto časti úlohy má študent k dispozícií topológiu so základnou konfiguráciou, ktorá zahŕňa nakonfigurované rozhrania a smerovací protokol OSPFv2. Ďalšia časť úlohy v sebe zahŕňa konfiguráciu IPv6 náležitostí ako samotná adresácia rozhraní smerovačov a konfiguráciu smerovacieho protokolu OSPFv3. Dôraz je ale kladený na konfiguráciu a princíp techniky MPLS vzhľadom na protokol IPv6. Protokol MPLS natívne nepodporuje IPv6 a jadro siete MPLS sa spolieha na protokol IPv4. Úloha obsahuje dva spôsoby, ktoré riešia problém konektivity IPv6 oblastí, ktoré sú oddelené MPLS sieťou. Ako prvý spôsob je v úlohe postup konfigurácie tunela typu 6to4. Druhý spôsob riešenia je konfigurácia protokolu MP–BGP v rámci MPLS smerovačov PE. Návody k laboratórnej úlohe sú v prílohe A.4 na strane 123.



Obr. 7.13: Topológia zadania laboratórnej úlohy 4.

8 ZÁVER

Táto diplomová práca obsahuje štyri laboratórne úlohy. Úlohy sú koncipované pre simulačné prostredie GNS3.

Prvá úloha sa zameriava na smerovacie protokoly OSPFv2 a OSPFv3. Obsahuje konfiguráciu autentizácie správ OSPF, konfiguráciu NAT–PT a posledná časť sa venuje tunelovacím technikám. V tejto časti sa nachádza konfigurácia tunelu typu GRE a 6to4.

Druhá úloha sa zameriava v prvej časti na konfiguráciu smerovacích protokolov EIGRP a EIGRPv6. Ďalej obsahuje konfiguráciu protokolov DHCP a DHCPv6. Protokol DHCPv6 sa konfiguruje ako stavový tak aj bezstavový. V poslednej časti úlohy je rozbor protokolu ICMP, pričom sa úloha primárne zameriava na protokol ICMPv6 kde je aj problematika fragmentovania správ.

Tretia úloha rieši otázku bezpečnosti v IPv6. Pôvodne mala úloha obsahovať aj zabezpečenie prístupovej vrstvy. Bohužiaľ modul EtherSwitch nepodporoval aplikovanie príkazov pre protokolovú sadu IPv6. Úloha takto obsahuje zabezpečenie smerovacieho protokolu OSPFv3, konfiguráciu IPsec tunelu, prístupových zoznamov a záver úlohy je venovaný stavovému Cisco IOS firewallu.

Celá štvrtá úloha sa zameriava na protokol MPLS. Úloha je rozdelená na niekoľko častí. Prvá úvodná časť sa zameriava na základnú konfiguráciu MPLS. V tejto časti sa študent v jednoduchosti zoznámi so základmi MPLS. Ďalšia časť popisuje dôležitosť parametru *Router ID* v rámci MPLS. Ďalej sa úloha venuje konfigurácií autentizácie MPLS. Posledná časť sa venuje protokolu MPLS v IPv6. Sú tu uvedené dve možné techniky využitia MPLS v IPv6. Jedna z nich je konfigurácia tunela 6to4 a druhá je použitie protokolu MP–BGP.

Všetky úlohy boli koncipované tak aby zachytávali rozdielnosť protokolov IPv4 a IPv6. Vo všetkých úlohách sa vyskytujú výpisy z prostredia IOS. V rámci simulačného programu GNS3 a s podporou programu Wireshark, určeného na zachytávanie sieťovej prevádzky, sa v úlohách vyskytuje množstvo výpisov a ilustrácií práve z uvedeného Wiresharku. Všetky uvedené výpisy majú slúžiť na zrozumiteľné a názorne vysvetlenie rôznych procesov v rámci sieťovej prevádzky. Na konci každej laboratórnej úlohy sa nachádza sada kontrolných otázok, ktoré majú overiť vedomosti študenta z danej problematiky. Otázky samozrejme vychádzajú z daných úloh. Každá úloha obsahuje zároveň aj samostatnú doplnkovú úlohu. Tak isto táto úloha vychádza z faktov, ktoré sú obsiahnuté v danej úlohe. Na základe informácií, ktoré sú obsiahnuté v jednotlivých úlohách by študent nemal mať problém zodpovedať kontrolné otázky a vypracovať samostatnú úlohu. Navrhnuté laboratórne úlohy sú stavané tak, aby ich študent bol schopný stihnúť úspešne vypracovať za približne dve a pol hodiny.

LITERATÚRA

[1] Adding hosts to your Topologies GNS3. *Graphical Network Simulator GNS3* [online]. 2014 [cit. 2014-10-16]. Dostupné z URL: <http://www.gns3.net/documentation/gns3/adding-hosts-to-your-topologies/>.

[2] BOOTP / DHCP options. *Network Sorcery* [online]. 2012 [cit. 2014-11-11]. Dostupné z URL: <http://www.networksorcery.com/enp/protocol/bootp/options.htm>.

[3] Cisco IOS Firewall - Products & Services - Cisco. *Cisco Systems, Inc* [online]. [online]. [cit. 2015-03-30]. Dostupné z URL: <http://www.cisco.com/c/en/us/products/security/ios-firewall/index.html>.

[4] Cisco IOS Packaging Customer Q&A: Cisco IOS Packaging. *Cisco Systems, Inc* [online]. [online]. [cit. 2014-10-29]. Dostupné z URL: <http://www.cisco.com/en/US/products/sw/iosswrel/ps5460/products_qanda_item09186a00801af2c6.shtml>.

[5] CONTA, A., S. DEERING a M. GUPTA, ED. RFC 2460. *Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification.* [online]. The Internet Society, 2006, 24 s. Dostupné z URL: <https://tools.ietf.org/html/rfc4443>.

[6] DEERING, S. a R. HINDEN. RFC 2460. *Internet Protocol, Version 6 (IPv6) Specification.* [online]. The Internet Society, 1998, 39 s. Dostupné z URL: <http://tools.ietf.org/html/rfc2460#page-4>.

[7] DHCPv6 Based IPv6 Access Services - Cisco. *Cisco Systems, Inc* [online]. 2011 [cit. 2014-10-24]. Dostupné z URL: <http://www.cisco.com/c/en/us/products/collateral/ios-nx-os-software/enterprise-ipv6-solution/whitepaper_c11-689821.html>.

[8] Differences in OSPFv3 from OSPFv2 - Knowledge Base. *Knowledge Base* [online]. [cit. 2014-12-02]. Dostupné z URL: <https://sites.google.com/site/amitsciscozone/home/important-tips/ipv6/differences-in-ospfv3-from-ospfv2>.

[9] Dynamips. *Graphical Network Simulator GNS3* [online]. 2014, [cit. 2014-10-07]. Dostupné z URL: <http://www.gns3.net/dynamips/>.

[10] GNS3 *Graphical Network Simulator GNS3* [online]. 2014, [cit. 2014-10-12]. Dostupné z URL: <http://www.gns3.net/>.

[11] GROSSETETE, Patrick. *Cisco IOS IPv6 Provider Edge Router (6PE) over MPLS* [online]. Cisco Systems, Inc, 2002, 36 s. [cit. 18.2.2015]. Dostupné z URL: <http://www.ipv6-tf.com.pt/documentos/geral/cisco/ipv6_IPV6overMPLS_Jul2002.pdf>.

[12] Hardware emulated by GNS3. *GNS3* [online]. [online]. 2014, 20.10.2014 [cit. 2014-10-25]. Dostupné z URL: <https://community.gns3.com/docs/DOC-1708>.

[13] HOGG, Scott a Eric VYNCKE. *IPv6 security.* [online]. Indianapolis: Cisco Press, 2009, xxi, 540 s. ISBN 978-1-58705-594-2.

[14] H3C - Products & Solutions - DHCP Introduction. *H3C Technologies Co., Limited* [online]. 2013 [cit. 2014-10-23]. Dostupné z URL: <http://www.h3c.com/portal/Products___Solutions/Technology/IPv4___IPv6_Services/Technology_Introduction/200701/195562_57_0.htm>.

[15] ICMPv4 - Internet Control Message Protocol for IPv4 - Sixscape Communications. HUGHES, Lawrence. *Home - Sixscape Communications* [online]. 2014 [cit. 2014-10-21]. Dostupné z URL: <http://www.sixscape.com/joomla/sixscape/index.php/technical-backgrounders/tcp-ip/ip-the-internet-protocol/ipv4-internet-protocol-version-4/icmpv4-internet-control-message-protocol-for-ipv4>.

[16] Implementing IPv6 over MPLS. *Cisco Systems, Inc* [online]. 2003, 1.5.2006 [cit. 2015-02-17]. Dostupné z URL: <http://www.cisco.com/c/en/us/td/docs/ios/ipv6/configuration/guide/12_4t/ipv6_12_4t_book/ip6-over_mpls.html>.

[17] Internet Control Message Protocol (ICMP) Parameters. *Internet Assigned Numbers Authority* [online]. 2013 19.4. [cit. 2014-10-21]. Dostupné z URL: <http://www.iana.org/assignments/icmp-parameters/icmp-parameters.xhtml>.

[18] Internet Control Message Protocol version 6 (ICMPv6) Parameters. *Internet Assigned Numbers Authority* [online]. [online]. 22.9.2014 [cit. 2014-10-26]. Dostupné z URL: <http://www.iana.org/assignments/icmpv6-parameters/icmpv6-parameters.xhtml>.

[19] IOS modes. *CMIT 265 Networking Essentials* [online]. 2014, [cit. 2014-10-08]. Dostupné z URL: <http://davidwills.net/cmit350/Images/ios_modes.png>.

[20] IP protokol. *Internet a Intranet: Velký průvodce protokoly* [online]. 2011 [cit. 2013-03-29]. Dostupné z URL: <http://zam.opf.slu.cz/botlik/CD-0x/5.html>.

[21] IPV4 Vs IPV6. *HTTP Debugger 6.5: HTTP Sniffer and Analyzer For Developers* [online]. 2014 [cit. 2014-11-11]. Dostupné z URL: <http://www.httpdebugger.com/articles/ipv4_vs_ipv6.html>.

[22] IPv6 Address Types In: *RIPE Network Coordination Centre: IPv6 Address Types* [online]. Január 2011, Január 2013 [cit. 2014-10-14]. Dostupné z URL: <https://blog.icann.org/wp-content/uploads/2010/07/ipv6-address-types.pdf>.

[23] IPv6 Training (Cisco) - How is EIGRPv6 Different From EIGRPv4?. *EzineArticles Submission* [online]. 2008 [cit. 2014-12-02]. Dostupné z URL: <http://ezinearticles.com/?IPv6-Training-(Cisco)---How-is-EIGRPv6-Different-From-EIGRPv4?&id=1357972>.

[24] JEŘÁBEK, Ing. Jan Ph.D. *Pokročilé komunikační techniky* [online]. 2014, [cit. 2014-10-13]. Dostupné z URL: <https://www.vutbr.cz/www_base/priloha.php?dpid=67088>.

[25] LAMMLE, Todd. *CCNA Cisco certified network associate study guide.* 6th ed. Indianapolis, Ind.: Wiley Pub., 2007, xxxix, 965 p. ISBN 978-047-0110-089

[26] MANRAL, Vishwas, Carlos PIGNATARO, Rajiv ASATI a Rajiv PAPNEJA. MPLS WORKING GROUP. *Updates to LDP for IPv6* [patent]. Updates to LDP for IPv6, draft-ietf-mpls-ldp-ipv6-16. Uděleno 2015. Zapsáno 11.2.215. Dostupné z URL: <https://tools.ietf.org/html/draft-ietf-mpls-ldp-ipv6-16>.

[27] Memory and CPU Usage. *Graphical Network Simulator GNS3* [online]. 2014, [cit. 2014-10-07]. Dostupné z URL: <http://www.gns3.net/documentation/gns3/memory-and-cpu-usage/>.

[28] MPLS. *MPLS Concepts.* [online]. Cisco Systems, Inc, 2002, 60 s.[cit. 2015-02-08]. <https://www.10gea.org/images/CISCO-MPLS-Concept.pdf>.

[29] NAT64 Technology: Connecting IPv6 and IPv4 Networks - Cisco. *Cisco Systems, Inc* [online]. 2012, Apríl 2012 [cit. 2014-10-12]. Dostupné z URL: <http://www.cisco.com/c/en/us/products/collateral/ios-nx-os-software/enterprise-ipv6-solution/white_paper_c11-676278.html>.

[30] ODOM, Wendell. *CCNP Route 642-902 official certification guide.* Indianapolis: Cisco Press, 2010, xxxiv, 730 s. ISBN 978-1-58720-253-7

[31] RANJBAR, Amir. *Troubleshooting and mainting cisco IP networks (TSHOOT) foundation learning guide: foundation learning for the CCNP TSHOOT 642-832.* 1st ed. Indianapolis: Cisco Press, 2010, xviii, 531 s. ISBN 978-1-58705-876-9.

[32] RFC 792. *Internet Control Message Protocol: Darpa Internet Program Protocol Specification.* Network Working Group, 1981. Dostupné z URL: <`https://www.ietf.org/rfc/rfc792.txt`>.

[33] RFC 3315. *Dynamic Host Configuration Protocol for IPv6 (DHCPv6).* The Internet Society, 2003.Dostupné z URL: <`http://www.networksorcery.com/enp/rfc/rfc3315.txt`>.

[34] RFC 5952. *A Recommendation for IPv6 Address Text Representation.* 2010. Dostupné z URL: <`http://tools.ietf.org/html/rfc5952`>.

[35] Switching simulation in GNS3. *Graphical Network Simulator GNS3* [online]. 2014, [cit. 2014-10-07]. Dostupné z URL: <`http://www.gns3.net/documentation/gns3/switching-simulation-in-gns3/`>.

[36] WELSH, Chris. *GNS3 Network Simulation Guide.* [online]. Livery Place: Packt Publishing, 2013. ISBN 978-1782160809.

[37] White Paper: Cisco IOS and NX-OS Software Reference Guide - Cisco Systems. *Cisco Systems, Inc* [online]. 2014, [cit. 2014-10-08]. Dostupné z URL: <`http://www.cisco.com/web/about/security/intelligence/ios-ref.html`>.

ZOZNAM SYMBOLOV, VELIČÍN A SKRATIEK

3DES	Trojitý šifrovací štandard pre dáta – Triple Data Encryption Standard
6to4	Pripojenie IPv6 domén cez IPv4 oblaky – Connection of IPv6 Domains via IPv4 Clouds
AHP	Autentizačný protokol hlavičiek – Authentication Header Protocol
AMD	Pokročilé mikro zariadenia – Advanced Micro Devices
ASIC	Aplikáciou špecifikovaný integrovaný obvod – Application Specific Integrated Circuit
ATM	Asynchrónny prenosový režim – Asynchronous Transfer Mode
BGP	Smerovací protokol – Border Gateway Protocol
CatOS	Catalyst operačný systém – Catalyst Operating System
CCNP	Cisco certifikovaný sieťový profesionál – Cisco Certified Network Professional
CE	Hranica zákazníka – Customer Edge
CEF	Technika smerovania paketov – Cisco Express Forwarding
CLI	Rozhranie príkazového riadku – Command Line Interface
CoS	Trieda služby – Class of Service
CRC	Cyklická redundantná kontrola – Cyclic Redundancy Check
DF	Nefragmentovať – Don't Fragment
DHCP	Protokol dynamickej konfigurácie koncového uzla – Dynamic Host Configuration Protocol
DH	Asymetrický šifrovací algoritmus – Diffie–Hellman
DNS	Systém názvov domén – Domain Name System
DSL	Digitálne účastnícke vedenie – Digital Subscriber Line
DUID	DHCP unikátny identifikátor – DHCP Unique Identifier
EIGRP	Smerovací protokol – Enhanced Interior Gateway Routing Protocol

EIGRPv6	Smerovací protokol verzia 6 – Enhanced Interior Gateway Routing Protocol version 6
ELSR	Typ smerovača v MPLS – Edge Label Switch Router
ESP	Zabezpečené zapúzdrovanie dát – Encapsulating Security Payload
EUI–64	Rozšírený unikátny identifikátor – Extended Unique Identifier
FW	Bezpečnostné zariadenie (Ohnivá brána) – Firewall
GNS3	Grafický sieťový simulátor 3 – Graphical Network Simulator 3
GRE	Generické smerovacie zapuzdrenie – Generic Routing Encapsulation
HDLC	Protokol linkovej vrstvy – High–Level Data Link Control
HTTP	Hypertextový prenosový protokol – Hypertext Transfer Protocol
HTTPS	Zabezpečený hypertextový prenosový protokol – Hypertext Transfer Protocol Secure
IA	Združenie identity – Identity Association
IAID	Identifikácia združenia identity – Identity Association Identification
iBGP	Interné susedstvo BGP smerovačov – Internal BGP
IBM	Medzinárodný podnik strojov – International Business Machines
ICMP	Protokol sieťovej vrstvy – Internet Control Message Protocol
ID	Identifikácia – Identification
IDS	Systém na detekciu vniknutia – Intrusion Detection System
IDP	Systém na detekciu a prevenciu vniknutia – Intrusion Detection and Prevention
IGP	Vnútorný smerovací protokol – Interior Gateway Protocol
IKE	Internetová výmena kľúčov – Internet Key Exchange
Intel	Integrovaná elektronika – Integrated Electronics
IOS	Sieťový operačný systém – Internetwork Operating System
IPsec	Internetový protokol bezpečnosti – IP Security

IPv4	Internetový protokol verzia 4 – Internet Protocol version 4
IPv6	Internetový protokol verzia 6 – Internet Protocol version 6
IPX	Sada sieťových protokolov pre Novell – Internetwork Packet Exchange
ISAKMP	Internetová bezpečnostná asociácia a protokol kľúčového manažmentu – Internet Security Association and Key Management Protocol
ISATAP	Vnútrostranný automaticky tunelovo adresný protokol – Intra–Site Automatic Tunnel Addressing Protocol
ISO	Medzinárodná organizácia pre štandardizáciu – International Organization for Standardization
ISP	Poskytovateľ internetových služieb – Internet Service Provider
LAN	Miestna sieť – Local Area Network
LDP	Distribučný protokol štítkov – Label Distribution Protocol
LFIB	Informačná tabuľka štítkov – Label Forwarding Information Base
LS	Stav linky – Link–State
LSA	Oznámenie stavu linky – Link–State Advertisement
LSR	Typ smerovača v MPLS – Label Switch Router
Mac OS	Operačný systém Macintosh – Macintosh Operating System
MCT	Manuálne konfigurované tunely – Manually Configured Tunnels
MD5	Hešovací algoritmus 5 – Message–Digest Algorithm 5
MF	Viac fragmentov – More Fragments
MIPS	Procesor bez automaticky spojeného zreťazeného spracovania – Microprocessor without Interlocked Pipeline Stages
MP–BGP	Viacprotokolový BGP – Multiprotocol BGP
MPLS	Viacprotokolové značkové prepínanie – Multiprotocol Label Switching
MS	Americká nadnárodná firma – Microsoft
NAT	Preklad sieťových adries – Network Address Translation

NDP	Protokol objavovania susedov – Neighbor Discovery Protocol
NVI	NAT virtuálne rozhranie – NAT Virtual Interface
NVRAM	Permanentná pamäť s náhodným prístupom – Non–Volatile Random Access Memory
OSI	Otvorené systémy vzájomnej komunikácie – Open Systems Interconnection
OSPFv2	Smerovací protokol verzia 2 – Open Shortest Path First version 2
OSPFv3	Smerovací protokol verzia 3 – Open Shortest Path First version 3
P	Poskytovateľ – Provider
PC	Osobný počítač – Personal Computer
PE	Hranica poskytovateľa – Provider Edge
PPP	Protokol linkovej vrstvy – Point–to–Point Protocol
QoS	Kvalita služieb – Quality of Service
RAM	Pamäť s náhodným prístupom – Random Access Memory
RA	Oznámenie smerovača – Router Advertisement
RS	Výzva smerovaču – Router Solicitation
RSA	Šifrovací algoritmus – Rivest–Shamir–Adleman
SCTP	Protokol riadenia prenosu streamu – Stream Control Transmission Protocol
SP	Poskytovateľ služieb – Service Provider
SPF	Algoritmus hľadania najkratšej cesty – Shortest Path First
SPI	Index bezpečnostného parametru – Security Parameter Index
SSH	Zabezpečený komunikačný protokol – Secure Shell
SSL	Vrstva bezpečných soketov – Secure Sockets Layer
TCP	Vysielací kontrolný protokol – Transmission Control Protocol
TDP	Distribučný protokol štítkov – Tag Distribution protocol

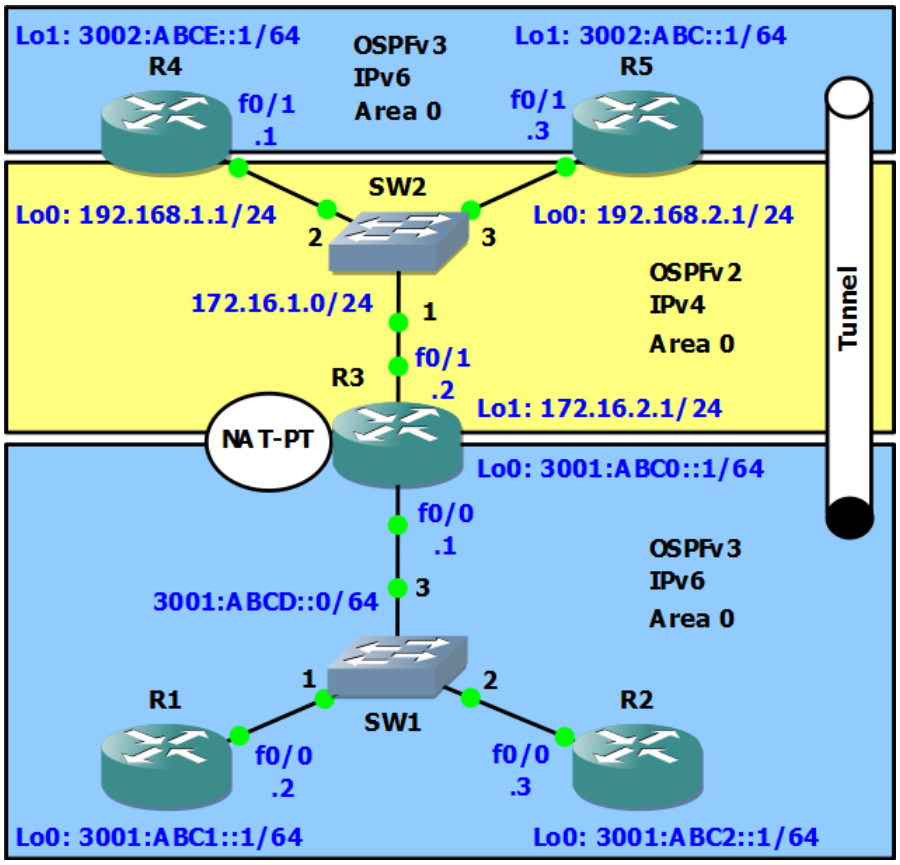
ToS	Typ služby – Type of Service
UDP	Používateľský datagramový protokol – User Datagram Protocol
VLAN	Virtuálna LAN – Virtual LAN
VoATM	Hlas prostredníctvom ATM – Voice over Asynchronous Transfer Mode
VoIP	Hlas prostredníctvom IP – Voice over IP
VoFR	Hlas prostredníctvom Frame Relay – Voice over Frame Relay
VPCS	Virtuálny simulátor PC – Virtual PC Simulator

ZOZNAM PRÍLOH

A PŘÍLOHA

A.1 Laboratorní úloha 1

A.1.1 Zapojení topologie sítě.



Obr. A.1: Zapojení topologie sítě

A.1.2 Hlavní cíle úlohy

- Konfigurace směrovacího protokolu OSPFv2
- Konfigurace směrovacího protokolu OSPFv3
- Konfigurace autentizace OSPF
- Konfigurace NAT–PT
- Konfigurace IPv6 tunelu

A.1.3 Teoretický úvod

Jedním z nejpoužívanějších IGP protokolů je právě protokol OSPF [31]. Pro verzi IPv4 nese označení OSPFv2 a pro verzi IPv6 to je OSPFv3. Techniky tunelování a překlad protokolů se řadí mezi tranzitní techniky v rámci migrace IPv4 a IPv6. Může nastat případ kdy uživatelé z IPv6 potřebují mít konektivitu na server, který nedisponuje možností IPv6. Na řešení tohoto problému můžeme využít techniku dynamického NAT–PT. V případě, že potřebujeme vytvořit trvalé spojení mezi dvěma IPv6 oblastmi, které jsou odděleny IPv4 oblastí, je vhodné použít GRE tunel. V rámci GRE tunelu můžeme použít různé IGP protokoly. V případě mobilních zařízení, kdy není přesně určena destinace uzlu, je výhodné použít dynamický tunel 6to4.

A.1.4 Úkol 1 – Konfigurace instance OSPFv2

a) Nejdřív si zprovozníme oblast sítě, která spadá pod protokolovou sadu IPv4. Na směrovačích R3, R4 a R5 nakonfigurujte IPv4 adresy včetně loopback rozhraní. Protokolem IPv6 se budeme zabývat později.

b) Nyní nakonfigurujeme směrovací protokol OSPFv2. Spustíme instanci OSPF s číslem procesu 1. Spuštění instance OSPF je možné dvěma způsoby. Jeden je klasicky za pomoci příkazu **router ospf** *číslo_instance*.

```
R3(config)#router ospf 1
```

Druhý způsob je v rámci konfiguračního režimu rozhraní, na kterém je zároveň dána síť, kterou chceme zařadit do směrovacího procesu.

```
R3(config)#interface fastEthernet0/1
R3(config-if)#ip ospf 1 area 0
```

Nyní došlo k přidělení 32 bitového identifikátoru *Router ID*. Můžeme si ověřit nastavení *Router ID* směrovače R3 příkazem:

```
R3#show ip ospf
Routing Process "ospf 1" with ID 172.16.2.1
<výstup zkrácen>
```

Nastavení směrovacího procesu proveďte i na směrovači R4 a R5.

c) Přistoupíme k přidání sítí do směrovacího procesu. Existují dva způsoby jak to můžeme udělat. První způsob je v režimu *config–router* přidat manuálně přímo připojené sítě.

```
R3(config)#router ospf 1
R3(config-router)#network adresa_sítě maska_sítě area id_oblasti
```

Druhou variantou je přímo v konfiguračním režimu rozhraní zadat příkaz pro přidání dané sítě do směrovacího procesu.

```
R3(config-if)#ip ospf id_procesu area id_oblasti
```

Nejdřív si ale na směrovači R4 zapneme výpis o navazování sousedství.

```
R4#debug ip ospf adj
```

Libovolným způsobem nakonfigurujte směrovače R3, R4 a R5, aby všechny IPv4 sítě byli součástí OSPF procesu. V konzolovém okně bychom pak měli vidět výpis o navazování sousedství, jak je vidět z ukázky výpisu:

```
R4#debug ip ospf adj
Nov 23 16:10:19.423: OSPF-1 ADJ    Fa0/1: Rcv DBD from 172.16.2.1 seq 0x1EB8 opt 0x52 flag
0x7 len 32  mtu 1500 state INIT
*Nov 23 16:10:19.423: OSPF-1 ADJ   Fa0/1: 2 Way Communication to 172.16.2.1, state 2WAY
<výstup zkrácen>
0x2 len 112  mtu 1500 state EXSTART
<výstup zkrácen>
0x0 len 32  mtu 1500 state EXCHANGE
<výstup zkrácen>
*Nov 23 16:10:19.763: OSPF-1 ADJ   Fa0/1: Synchronized with 172.16.2.1, state FULL
*Nov 23 16:10:19.767: OSPF-5-ADJCHG: Process 1, Nbr 172.16.2.1 on FastEthernet0/1 from
LOADING to FULL, Loading Done
<výstup zkrácen>
```

d) Zobrazíme si výpis ze směrovací tabulky směrovače R3.

```
R3#show ip route
   172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks
C     172.16.1.0/24 is directly connected, FastEthernet0/1
L     172.16.1.2/32 is directly connected, FastEthernet0/1
C     172.16.2.0/24 is directly connected, Loopback1
L     172.16.2.1/32 is directly connected, Loopback1
   192.168.1.0/32 is subnetted, 1 subnets
O 192.168.1.1 [110/2] via 172.16.1.1,00:00:11,FastEthernet0/1
   192.168.2.0/32 is subnetted, 1 subnets
O 192.168.2.1 [110/2] via 172.16.1.3,00:00:01,FastEthernet0/1
```

Z tabulky je vidět, že loopback sítě jsou přenášeny s maskou /32. Příkazem **ip ospf network point–to–point** nastavte na všech loopback rozhraních správný typ sítě. Následuje ukázka konfigurace tohoto parametru.

```
R3(config)#interface loopback0
R3(config-if)#ip ospf network point-to-point
```

Po opětovném zobrazení směrovací tabulky vidíme, že sítě se přenášejí se správnou maskou.

```
R3#show ip route
   172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks
C     172.16.1.0/24 is directly connected, FastEthernet0/1
L     172.16.1.2/32 is directly connected, FastEthernet0/1
C     172.16.2.0/24 is directly connected, Loopback1
L     172.16.2.1/32 is directly connected, Loopback1
O 192.168.1.0/24 [110/2] via 172.16.1.1,00:00:16,FastEthernet0/1
O 192.168.2.0/24 [110/2] via 172.16.1.3,00:00:16,FastEthernet0/1
```

Samostatně s příkazem **ping** otestujte konektivitu sítě. Celá IPv4 oblast by měla mít plnou konektivitu.

A.1.5 Úkol 2 – Konfigurace instance OSPFv3

a) Teď si nakonfigurujeme IP adresy všech rozhraní v oblasti IPv6. Nejdřív si ale na rozhraní FastEthernet0/1 směrovače R1 vyzkoušíme příkaz:

```
R1(config)#interface FastEthernet0/1
R1(config-if)#ipv6 enable
```

Po aplikování příkazu **ipv6 enable** si zobrazte IP adresy rozhraní příkazem:

```
R1#show ipv6 interface brief
FastEthernet0/0        [administratively down/down]
    unassigned
FastEthernet0/1        [administratively down/down]
    FE80::C806:7FF:FE2C:6
```

Zamyslete se, co je to za adresu a jak přesně vznikla. Konfigurace rozhraní pro verzi IPv6 probíhá obdobně jak u verzi IPv4. Například pro směrovač R1 bude konfigurace vypadat následovně:

```
R1(config-if)#ipv6 address adresa_sítě/prefix
```

Samostatně nakonfigurujte IP adresaci na všech směrovačích včetně loopback rozhraní.

b) Dále musíme aktivovat v konfiguračním režimu IPv6 směrování příkazem:

```
R1(config)#ipv6 unicast-routing
```

Tento krok proveďte na všech potřebných směrovačích. Bez tohoto parametru by nebylo možné spustit IPv6 směrování na daném zařízení.

c) Obdobně jako u OSPFv2 i v tomto případě máme na výběr dva způsoby spuštění OSPFv3 instance. My si pro aktivovaní OSPFv3 procesu zadáme do konfiguračního režimu následující příkaz:

```
R1(config)#ipv6 router ospf 1
```

Číslo 1 představuje identifikační číslo procesu směrovacího protokolu OSPFv3 obdobně jako to bylo u OSPFv2. Po tom, co potvrdíme tenhle příkaz, na výstupu konzole se nám zobrazí hláška znázorněna níže.

```
R1(config)#ipv6 router ospf 1
R1(config-rtr)#
*Oct 11 21:02:04.451: %OSPFv3-4-NORTRID: Process OSPFv3-1-IPv6
could not pick a router-id, please configure manually
R1(config-rtr)#
```

Router ID je vždy 32 bitová hodnota, což představuje IPv4 adresu. Zatím jsme na rozhraní směrovače R1 nenastavovali žádnou IPv4 adresu, což ani nemusíme. Nastavíme přímo *Router ID* v subkonfiguračním režimu směrovače.

```
R1(config)#ipv6 router ospf 1
R1(config-rtr)#router-id 1.1.1.1
```

Při dotazu o *Router ID* na směrovači R2 nakonfigurujte parametr 2.2.2.2.
Poznámka: Všimněte si, že konfigurační režim směrovacího procesu OSPFv3 je naznačen jako *config–rtr*, zatím co u OSPFv2 to bylo *config–router*.

d) Spustíme debug výpis ohledně sestavování sousedů příkazem **debug ipv6 ospf adj** na směrovači R3. Přidáme jednotlivé sítě do směrovacího procesu. Tady již nemáme na výběr možnost přidání sítě přes subkonfigurační režim směrovacího procesu. Použijeme výše zmíněný příkaz pro verzi IPv6.

```
R1(config-if)#ipv6 ospf id_procesu area id_oblasti
```

Po tomto se podíváme na debug výpis o navazování sousedství. Nasleduje ukázka výpisu.

```
R3#debug ipv6 ospf adj
*Jan 26 13:39:04.595: OSPFv3-1-IPv6 ADJ   Fa0/0: Cannot see ourself in hello from 2.2.2.2,
 state INIT
<výstup zkrácen>
*Jan 26 13:39:04.771: OSPFv3-1-IPv6 ADJ   Fa0/0: 2 Way Communication to 2.2.2.2, state 2WAY
<výstup zkrácen>
*Jan 26 13:39:04.811: OSPFv3-1-IPv6 ADJ   Fa0/0: Rcv DBD from 2.2.2.2 seq 0x37D8B805 opt
0x0013 flag 0x7 len 28  mtu 1500 state EXSTART
<výstup zkrácen>
*Jan 26 13:39:04.907: OSPFv3-1-IPv6 ADJ   Fa0/0: Rcv DBD from 2.2.2.2 seq 0x16FD211 opt
0x0013 flag 0x2 len 48  mtu 1500 state EXSTART
<výstup zkrácen>
*Jan 26 13:39:05.095: OSPFv3-1-IPv6 ADJ   Fa0/0: Rcv DBD from 2.2.2.2 seq 0x16FD212 opt
0x0013 flag 0x0 len 28  mtu 1500 state EXCHANGE
<výstup zkrácen>
*Jan 26 13:39:05.095: OSPFv3-1-IPv6 ADJ   Fa0/0: Synchronized with 2.2.2.2, state FULL
*Jan 26 13:39:05.099: %OSPFv3-5-ADJCHG: Process 1, Nbr 2.2.2.2 on FastEthernet0/0
from LOADING to FULL, Loading Done
```

Z výpisu se dá vyčíst, že identifikace sousedství probíhá prostřednictvím identifikátoru *Router ID*.

e) Správnost konfigurace OSPFv3 protokolu ověříme následujícím příkazem:

```
R3#show ipv6 route
C   3001:ABC0::/64 [0/0]
     via Loopback0, directly connected
L   3001:ABC0::1/128 [0/0]
     via Loopback0, receive
O 3001:ABC1::1/128 [110/1]
     via FE80::C801:1EFF:FE58:8, FastEthernet0/0
O 3001:ABC2::1/128 [110/1]
     via FE80::C802:11FF:FEA0:8, FastEthernet0/0
C   3001:ABCD::/64 [0/0]
     via FastEthernet0/0, directly connected
L   3001:ABCD::1/128 [0/0]
     via FastEthernet0/0, receive
L   FF00::/8 [0/0]
     via Null0, receive
```

Můžeme si všimnout, že adresy se přenášejí se špatnou maskou. Na základě dřívějších zkušeností samostatně nakonfigurujte potřebné parametry, aby přenášena maska byla správná. Samostatně ověřte výsledek vaši konfigurace.

f) Zobrazíme si tabulky OSPF databáze pro směrovače R4 a R1 příslušným příkazem.

```
R4#show ip ospf database
                Router Link States (Area 0)
Link ID         ADV Router      Age         Seq#       Checksum Link count
172.16.2.1      172.16.2.1      25          0x80000002 0x00FF62 2
192.168.1.1     192.168.1.1     20          0x80000002 0x00C09F 2
192.168.2.1     192.168.2.1     30          0x80000002 0x00BF9D 2
                Net Link States (Area 0)
Link ID         ADV Router      Age         Seq#       Checksum
172.16.1.1      192.168.168.5   24          0x80000002 0x004DCE
```

Z výpisu OSPFv3 databázi vidíme dva nové typy LSA zpráv. Link State LSA (typ 8) se používá k získání informací o linkových lokálních adresách a seznamu IPv6 adres na dané lince. Intra–area Prefix LSA (typ 9) obsahuje prefixi pro stub a tranzitní sítě.

```
R1#show ipv6 ospf database
<výstup zkrácen>
                Link (Type-8) Link States (Area 0)
ADV Router       Age         Seq#        Link ID    Interface
 1.1.1.1         131         0x80000004  10         Lo0
 1.1.1.1         131         0x80000004  3          Fa0/0
 2.2.2.2         1452        0x80000003  3          Fa0/0
 172.16.2.1      217         0x80000005  3          Fa0/0
                Intra Area Prefix Link States (Area 0)
ADV Router       Age         Seq#        Link ID    Ref-lstype  Ref-LSID
 1.1.1.1         131         0x80000005  0          0x2001      0
 172.16.2.1      217         0x80000006  0          0x2001      0
 172.16.2.1      217         0x80000005  3072       0x2002      3
```

g) Spusťte zachytávání provozu na směrovači R1 a R4 pomocí Wireshark. V konzolovém okně těchto směrovačů zadejte nasledující příkaz.

```
R1#clear ipv6 ospf process
```

Obrázky A.2 a A.3 ilustrují zachycený provoz na směrovačích R1 a R4. V protokolu OSPFv2 se skupinová adresa 224.0.0.5 používá k zasílání hello zpráv a LS (Link–State) aktualizací. Adresa 224.0.0.6 se používá v rámci komunikace DR/BDR. U OSPFv3 se k tomuto používají adresy FF02::6 a FF02::5. Všimněte si, že poslední bajt adresy zůstal stejný v obou verzích.

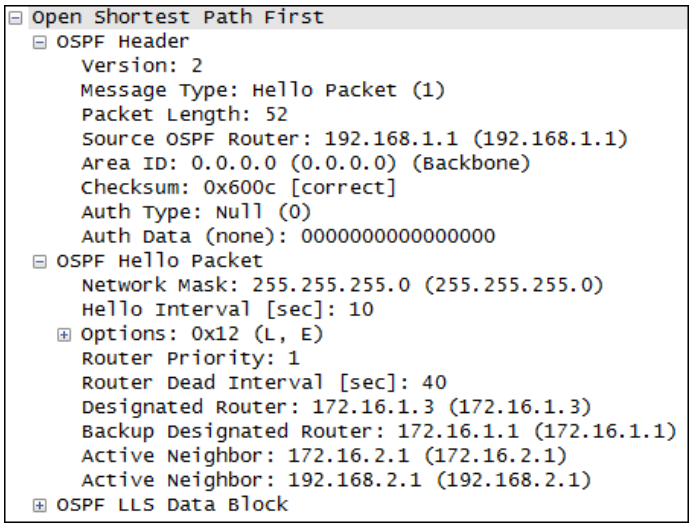
Source	Destination	Protocol	Length	Info
172.16.1.3	172.16.1.1	OSPF	118	DB Description
172.16.1.2	224.0.0.6	OSPF	126	LS Update
172.16.1.3	224.0.0.5	OSPF	126	LS Update
172.16.1.3	224.0.0.5	OSPF	158	LS Update
172.16.1.2	224.0.0.5	OSPF	138	Hello Packet

Obr. A.2: Zachycený provoz na směrovači R4.

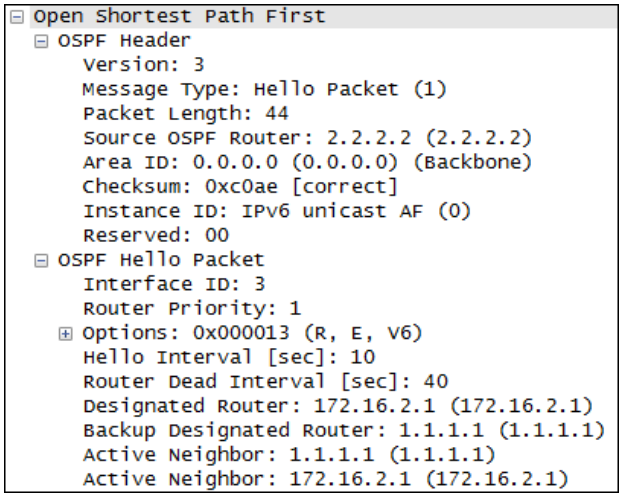
Source	Destination	Protocol	Length	Info
fe80::c801:1eff:fe58:8	ff02::6	OSPF	154	LS Update
fe80::c803:14ff:feb4:8	ff02::5	OSPF	154	LS Update
fe80::c802:11ff:fea0:8	ff02::5	OSPF	174	LS Acknowledge
fe80::c803:14ff:feb4:8	ff02::5	OSPF	134	LS Update
fe80::c801:1eff:fe58:8	ff02::5	OSPF	122	Hello Packet

Obr. A.3: Zachycený provoz na směrovači R1.

Obrázek A.4 ilustruje zprávu OSPFv2, zatímco na obrázku A.5 je znázorněna správa OSPFv3. Samostatně porovnejte zobrazené informace. OSPFv2 hello paket nese informace mimo jiné o 8bitovém poli *Option* a 32bitovém poli *Dead Interval*. U OSPFv3 je to 24bitové pole *Option* a 16bitové pole *Dead Interval*.



Obr. A.4: Hlavička paketu OSPFv2.



Obr. A.5: Hlavička paketu OSPFv3.

h) Na směrovači R3 si příslušným debug příkazem necháme vypsat přicházející OSPF pakety. Zamyslete se nad významem jednotlivých atributů.

```
R3#debug ip ospf packet
OSPF-1 PAK  : rcv. v:2 t:1 l:52 rid:192.168.2.1 aid:0.0.0.0
chk:600C aut:0 auk: from FastEthernet0/1
```

```
R3#debug ipv6 ospf packet
OSPFv3-1-IPv6 PAK   Fa0/0: rcv. v:3 t:1 l:44 rid:2.2.2.2
OSPFv3-1-IPv6 PAK   Fa0/0: aid:0.0.0.0 chk:C0AE inst:0 from
FastEthernet0/0
```

Atribut *v* značí verzi OSPF protokolu. Atribut *t* značí typ zprávy a v tomto případě je to 1 což představuje hello zprávu. Atribut *l* představuje délku zprávy. Atribut *aut* představuje autentizaci, přičemž hodnota 0 indikuje, že se autentizace nevyužívá.

A.1.6 Úkol 3 – Konfigurace autentizace OSPF

a) Nejprve nakonfigurujeme autentizaci v rámci oblasti OSPFv2. Na směrovačích R3, R4 a R5 proveďte následující příkazy:

```
R3(config)#router ospf 1
R3(config-router)#area 0 authentication message-digest
R3(config-router)#exit
R3(config)#interface fastEthernet0/1
R3(config-if)#ip ospf message-digest-key 1 md5 cisco
```

Zobrazíme si stav autentizace na daném rozhraní.

```
R3#show ip ospf interface fastEthernet0/1
<výstup zkrácen>
Message digest authentication enabled
    Youngest key id is 1
```

S příslušným debug příkazem si ověříme zasílání autentizovaných hello zpráv.

```
R3#debug ip ospf adj
*Nov 23 21:17:14.751: OSPF-1 ADJ   Fa0/1: Send with youngest Key 1
```

Stejný stav by měly interpretovat výpisy ze směrovačů R4 a R5. Tento stav samostatně zkontrolujte.

b) Dále nakonfigurujeme autentizaci v rámci oblasti OSPFv3. Na směrovačích R3, R1 a R2 proveďte následující příkazy:

```
R3(config)#interface fastEthernet0/0
R3(config-if)#ipv6 ospf authentication ipsec spi 256
md5 12345678901234567890123456789012
```

Zobrazíme si stav autentizace na daném rozhraní. Následuje zkrácený výpis.

```
R3#show ipv6 ospf interface fastEthernet0/0
<výstup zkrácen>
MD5 authentication SPI 256, secure socket UP (errors: 0)
<výstup zkrácen>
```

Stejný výpis bychom měli obdržet na směrovačích R1 a R2. Tento stav samostatně zkontrolujte.

c) Na směrovačích R1 a R4 spustíme zachytávání provozu. Najděte ve zprávách údaje, které charakterizují autentizaci zpráv. Nejpodstatnější rozdíl problematiky autentizace v rámci protokolu OSPFv2 a OPSFv3 je ve způsobu řešení. Zatímco u OSPFv2 se autentizuje pomocí zahešovaných zpráv, u OSPFv3 se využívá IPsec protokol. Obrázek A.6 ilustruje řešení autentizace na směrovači R4. Obrázek A.7 ilustruje řešení autentizace na směrovači R1.

```
⊞ Frame 3: 134 bytes on wire (1072 bits), 134 bytes captured (1072 bits) on interface 0
⊞ Ethernet II, Src: ca:03:14:b4:00:06 (ca:03:14:b4:00:06), Dst: IPv4mcast_05 (01:00:5e:00:00:05)
⊞ Internet Protocol Version 4, Src: 172.16.1.2 (172.16.1.2), Dst: 224.0.0.5 (224.0.0.5)
⊟ Open Shortest Path First
  ⊟ OSPF Header
      Version: 2
      Message Type: Hello Packet (1)
      Packet Length: 48
      Source OSPF Router: 172.16.2.1 (172.16.2.1)
      Area ID: 0.0.0.0 (0.0.0.0) (Backbone)
      Checksum: 0x0000 (None)
      Auth Type: Cryptographic (2)
      Auth Crypt Key id: 1
      Auth Crypt Data Length: 16
      Auth Crypt Sequence Number: 1422737329
      Auth Crypt Data: 15b91492e03e68b3db98e76054f02465
  ⊞ OSPF Hello Packet
  ⊞ OSPF LLS Data Block
```

Obr. A.6: Ukázka autentizace u OSPFv2.

```
⊟ Internet Protocol Version 6, Src: fe80::c801:1eff:fe58:8 (fe80::c801:1eff:fe58:8),
  ⊞ 0110 .... = Version: 6
  ⊞ .... 1110 0000 .... .... .... .... .... = Traffic class: 0x000000e0
    .... .... .... 0000 0000 0000 0000 0000 = Flowlabel: 0x00000000
    Payload length: 68
    Next header: Authentication Header (51)
    Hop limit: 1
    Source: fe80::c801:1eff:fe58:8 (fe80::c801:1eff:fe58:8)
    Destination: ff02::5 (ff02::5)
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
  ⊟ Authentication Header
      Next Header: OSPF IGP (0x59)
      Length: 24
      AH SPI: 0x00000100
      AH Sequence: 249
      AH ICV: 6bed3695c7cc92f887b7358d
⊞ Open Shortest Path First
```

Obr. A.7: Ukázka autentizace u OSPFv3.

A.1.7 Úkol 4 – Konfigurace NAT–PT

a) Veškerá nastavení se budou týkat směrovače R3. Jako první vytvoříme přístupový seznam, ve kterém definujeme konkrétní adresný rozsah, který se bude moci podléhat překladu adres.

```
R3(config)#ipv6 access-list NAT-PT-ACL
R3(config-ipv6-acl)#permit ipv6 3001::/16 any
```

b) Dále v konfiguračním režimu vytvoříme adresní rozsah (pool), na který se budou zdrojové IPv6 adresy překládat.

```
R3(config)#ipv6 nat v6v4 pool NAT_POOL_IPV4 172.16.2.10
172.16.2.20 prefix-length 24
```

c) Dalším krokem je namapovat IPv6 adresu 2001::C0A8:101 na IPv4 adresu 192.168.1.1, která bude reprezentovat v našem případě fiktivní server, na který se chceme dotazovat z IPv6 oblasti.

```
R3(config)#ipv6 nat v4v6 source 192.168.1.1 2001::C0A8:101
```

d) Nakonfigurujeme překlad adres z IPv6 na IPv4 následujícím příkazem.

```
R3(config)#ipv6 nat v6v4 source list NAT-PT-ACL pool
NAT_POOL_IPV4
```

e) Nakonfigurujeme prefix, který bude reprezentovat vnější síť.

```
R3(config)#ipv6 nat prefix 2001::/96
```

f) Ještě nám zbývá spustit NAT službu na daných rozhraních směrovače R3. Zároveň aktivujeme funkci IPv6 na rozhraní fastEthernet0/1.

```
R3(config)#interface range fastEthernet0/0, fastEthernet0/1
R3(config-if-range)#ipv6 nat
R3(config-if-range)#ipv6 enable
```

g) Ze směrovače R1 se příkazem **ping** budeme dotazovat na adresu 192.168.1.1 prostřednictvím adresy 2001::C0A8:101.

```
R1#ping 2001::C0A8:101
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001::C0A8:101, timeout is
2 seconds:
% No valid route for destination
Success rate is 0 percent (0/1)
```

Z výše uvedeného výpisu vyplývá, že směrovač R1 nezná cestu k cílové IPv6 adrese. Samostatně si zobrazte směrovací tabulku pro směrovač R1 příkazem **show ipv6 route**. Následně si zobrazíme směrovací tabulku pro směrovač R3.

```
R3#show ipv6 route
<výstup zkrácen>
C   2001::/96 [0/0]
     via NVI0, directly connected
<výstup zkrácen>
```

Příkazem **ipv6 nat prefix** *2001::/96* se vytvořil NVI (Nat Virtual Interface) a figuruje jako přímo připojená síť. Tuto síť je třeba redistribuovat v rámci OSPFv3 aby všechny směrovače byly schopny komunikace s touto sítí.

```
R3(config)#ipv6 router ospf 1
R3(config-rtr)#redistribute connected metric 20
```

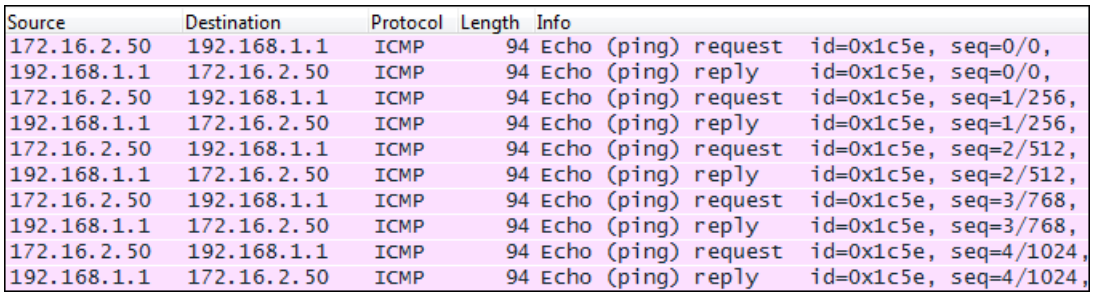
Samostatně ověřte dostupnost sítě 2001::/96 na směrovačích R1 a R2. Příkazem **ping** se několikrát dotazujte na adresu 2001:C0A8:101. Následuje výpis ze směrovače R1.

```
R1#ping 2001::C0A8:101
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001::C0A8:101, timeout is
2 seconds:
!!!!.
Success rate is 80 percent (4/5), round-trip min/avg/max =
56/67/84 ms
```

Můžete si všimnout, že ne všechny dotazy jsou úspěšné. Je to zapříčiněno funkcí CEF (Cisco Express Forwarding). NAT–PT nepodporuje tuto funkci. Toto je jeden z důvodů, proč je tehnika NAT–PT považována za zastaralou. V rámci úlohy si dočasně deaktivujeme funkci CEF. V ostrém provozu se toto nedoporučuje, protože by to mělo vážný dopad na výkonnost sítě.

```
R3(config)#no ipv6 cef
R3(config)#no ip cef
```

h) Na směrovači R3 spustíme příkazem **debug ipv6 nat detailed** výpisy ohledně překladu adres a následně ze směrovače R1 se příkazem **ping** budeme dotazovat na adresu 2001::C0A8:101. Zároveň spustíme zachytávání provozu na směrovači R3 rozhraní fastEthernet0/1. Obrázek A.8 ilustruje zachycený ICMP provoz po překladu adres na daném rozhraní směrovače. V rámci debug výpisu bychom měli být schopni zachytit výpis jako je znázorněn níže. Výpis si samostatně prostudujte.



Source	Destination	Protocol	Length	Info
172.16.2.50	192.168.1.1	ICMP	94	Echo (ping) request id=0x1c5e, seq=0/0,
192.168.1.1	172.16.2.50	ICMP	94	Echo (ping) reply id=0x1c5e, seq=0/0,
172.16.2.50	192.168.1.1	ICMP	94	Echo (ping) request id=0x1c5e, seq=1/256,
192.168.1.1	172.16.2.50	ICMP	94	Echo (ping) reply id=0x1c5e, seq=1/256,
172.16.2.50	192.168.1.1	ICMP	94	Echo (ping) request id=0x1c5e, seq=2/512,
192.168.1.1	172.16.2.50	ICMP	94	Echo (ping) reply id=0x1c5e, seq=2/512,
172.16.2.50	192.168.1.1	ICMP	94	Echo (ping) request id=0x1c5e, seq=3/768,
192.168.1.1	172.16.2.50	ICMP	94	Echo (ping) reply id=0x1c5e, seq=3/768,
172.16.2.50	192.168.1.1	ICMP	94	Echo (ping) request id=0x1c5e, seq=4/1024,
192.168.1.1	172.16.2.50	ICMP	94	Echo (ping) reply id=0x1c5e, seq=4/1024,

Obr. A.8: Zachycený ICMP provoz po překladu adres.

```
R3#debug ipv6 nat detailed
*Jan 27 14:36:17.371: IPv6 NAT: Found prefix 2001::/96
*Jan 27 14:36:18.339: IPv6 NAT: IPv6->IPv4:
                  src (3001:ABCD::2 -> 172.16.2.10)
                  dst (2001::C0A8:101 -> 192.168.1.1)
<výstup zkrácen>
*Jan 27 14:36:18.343: IPv6 NAT: IPv6->IPv4: icmp src (3001:ABCD::2)
-> (172.16.2.10), dst (2001::C0A8:101) -> (192.168.1.1)

*Jan 27 14:36:18.443: IPv6 NAT: Found prefix 2001::/96
*Jan 27 14:36:18.443: IPv6 NAT: IPv4->IPv6:
                  src (192.168.1.1 -> 2001::C0A8:101)
                  dst (172.16.2.10 -> 3001:ABCD::2)
<výstup zkrácen>
```

Na směrovači R3 si zobrazíme statistiky ohledně překladu NAT–PT následujícím příkazem.

```
R3#show ipv6 nat statistics
Total active translations: 3 (1 static, 2 dynamic; 1 extended)
NAT-PT interfaces:
  FastEthernet0/0, FastEthernet0/1, NVI0
Hits: 0  Misses: 0
Expired translations: 26
```

Z výše uvedeného výpisu si všimněte, že na rozdíl od překladu adres v rámci IPv4, ve verzi pro IPv6 se zde nachází v seznamu rozhraní mimo jiné i NVI. Dále si zobrazíme záznamy o překladu adres. Měli byste obdržet výpis jako je znázorněn níže.

```
R3#show ipv6 nat translations
Prot  IPv4 source              IPv6 source
      IPv4 destination         IPv6 destination
---   ---                      ---
      192.168.1.1              2001::C0A8:101
icmp  172.16.2.10,7564         3001:ABCD::2,7564
      192.168.1.1,7564         2001::C0A8:101,7564
---   172.16.2.10              3001:ABCD::2
```

Pokud si na směrovači R4 spustíme příkazem **debug ip icmp** výpisy ohledně ICMP protokolu a následně na to se budeme ze směrovače R1 dotazovat na adresu 2001::C0A8:101 měli bychom obdržet následující výpis.

```
R4#debug ip icmp
ICMP packet debugging is on
ICMP: echo reply sent, src 192.168.1.1, dst 172.16.2.10,
topology BASE, dscp 0 topoid 0
```

Otázka: Zamyslete se nad tím, co by se stalo v případě, kdybychom v rámci *NAT_POOL_IPV4* namísto adresního rozsahu 17.16.2.10 172.16.2.20 použili například roszah 10.10.10.10 10.10.10.20.

A.1.8 Úkol 5 – Konfigurace IPv6 tunelu GRE

a) Jako první, na směrovači R5 vytvoříme tunelové rozhraní a přidělíme mu IPv6 adresu.

```
R5(config)#interface Tunnel1
R5(config-if)#ipv6 address 2001:A:B:C::/64 eui-64
```

V tomto případě jsme využili techniku EUI–64. Pomocí této techniky se klientská část adresy odvodí z MAC adresy rozhraní. Příkazem **show ipv6 interface brief** si samostatně zobrazte adresy rozhraní vytvořeného tunelu.

b) Dále nastavíme zdroj tunelu jako rozhraní fastEthernet0/1 a nastavíme zapouzdřování GRE přes IPv4.

```
R5(config-if)#tunnel source fastEthernet0/1
R5(config-if)#tunnel mode gre ip
```

Cisco zařízení používají v rámci tunelu implicitně zapouzdřování GRE. V tomto případě je příkaz **tunnel mode gre ip** jen kvůli názornosti.

c) Předešlé kroky udělejte i na směrovači R3. V rámci tunelu použijeme směrovací protokol OSPFv3.

```
R3(config-if)#ipv6 ospf 1 area 0
R5(config-if)#ipv6 ospf 1 area 0
```

d) Jako poslední věc nastavíme cílovou adresu tunelu na obou směrovačích.

```
R3(config-if)#tunnel destination 172.16.1.3
R5(config-if)#tunnel destination 172.16.1.2
```

e) Příkazmi **show interfaces tunnel1** a **show ipv6 interfaces tunnel1** si zobrazíme bližší informace o vytvořeném tunely na směrovači R3.

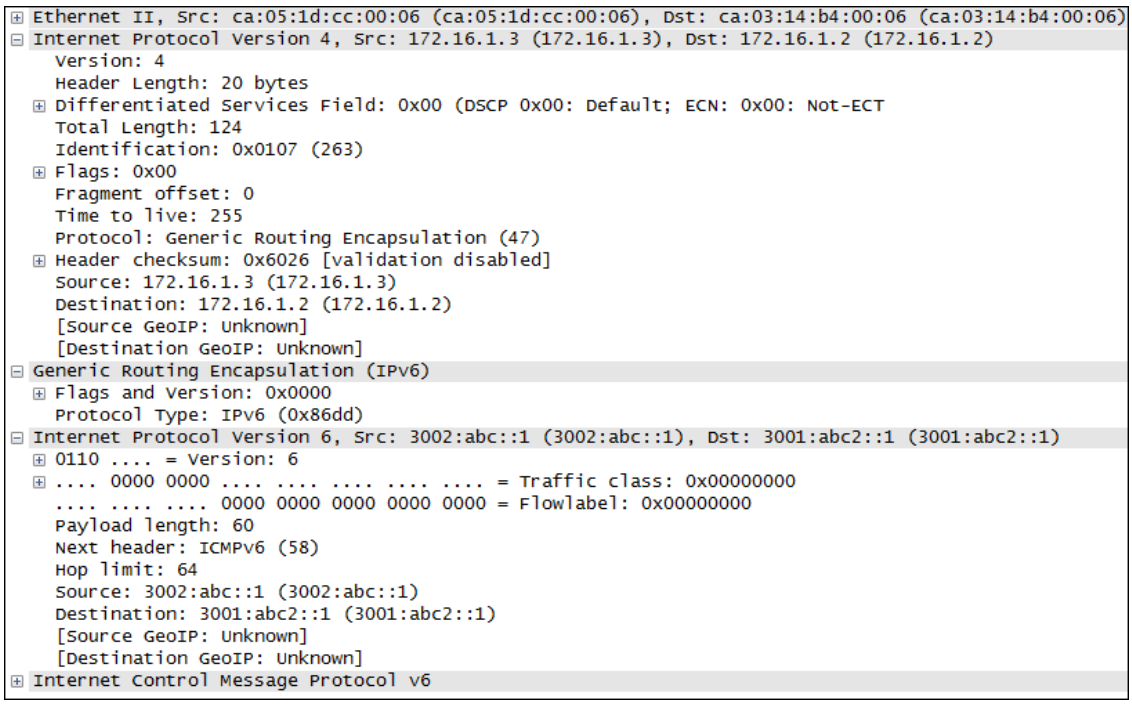
```
R3#show interfaces tunnel1
Tunnel1 is up, line protocol is up
  Hardware is Tunnel
  MTU 17916 bytes, BW 100 Kbit/sec, DLY 50000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation TUNNEL, loopback not set
  Keepalive not set
  Tunnel source 172.16.1.2 (FastEthernet0/1), destination 172.16.1.3
   Tunnel Subblocks:
      src-track:
         Tunnel1 source tracking subblock associated with
         FastEthernet0/1
          Set of tunnels with source FastEthernet0/1, 1 member
          (includes iterators), on interface <OK>
  Tunnel protocol/transport GRE/IP
    Key disabled, sequencing disabled
    Checksumming of packets disabled
  Tunnel TTL 255
  Tunnel transport MTU 1476 bytes
<výstup zkrácen>
```

```
R3#show ipv6 interface tunnel1
Tunnel1 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::C803:14FF:FEB4:8
  No Virtual link-local address(es):
  Global unicast address(es):
    2001:A:B:C:C803:14FF:FEB4:8, subnet is 2001:A:B:C::/64 [EUI]
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::5
    FF02::1:FFB4:8
  MTU is 1476 bytes
<výstup zkrácen>
```

f) Na směrovači R5 rozhraní fastEthernet0/1 si spustíme zachytávání provozu. Ze směrovače R5 se budeme dotazovat prostřednictvím příkazu **ping** na cílovou adresu 3001:ABC2::1, přičemž zdrojová adresa bude loopback1.

```
R5#ping 3001:abc2::1 source loopback1
```

V zachyceném provozu si prohlédněte první ping dotaz. Obrázek A.9 ilustruje zapouzdření ICMP paketu v rámci tunelu. V hlavičce protokolu IPv4 je informace v poli *Protocol* s číslem 47, která indikuje GRE zapouzdření. Hlavička GRE v poli *Protocol* nese informaci o typu zapouzdřeného protokolu. Hlavička IPv6 protokolu v poli *Next header* nese informaci s číslem 58 o použitém protokolu ICMP.



Obr. A.9: Zapouzdření ICMP paketu v rámci přechodu GRE tunelem

A.1.9 Úkol 6 – Konfigurace IPv6 tunelu 6to4

a) Jako první, na směrovači R4 vytvoříme tunelové rozhraní a přidělíme mu IPv6 adresu.

```
R4(config)#interface Tunnel0
R4(config-if)#ipv6 address 2002:AC10:101::/48
```

Druhý a třetí kvartál v sobě nese hexadecimální hodnotu IPv4 adresy rozhraní fastEthernet0/1. IPv4 adresa 172.16.1.1 má po převodu tvar AC10:101. Adresa tohoto rozhraní bude v dalším kroku použita jako zdrojová adresa tunelu.

b) Dále nastavíme zdroj tunelu jako adresu rozhraní fastEthernet0/1 a nastavíme režim tunelu jako 6to4.

```
R4(config-if)#tunnel source 172.16.1.1
R4(config-if)#tunnel mode ipv6ip 6to4
```

c) Obdobně nakonfigurujte směrovač R3. Potřebujeme nastavit statické směrování z důvodu, že tunel 6to4 nepodporuje IGP protokoly jako jsou OSPF nebo EIGRP. Tyto protokoly totiž pracují se skupinovými adresami v rámci udržování sousedství. Skupinové adresy tento typ tunelu nepodporuje. Statické směrování na směrovačích R3 a R4 nastavíme následovně.

```
R3(config)#ipv6 route 2002:AC10:101::/48 Tunnel0
R3(config)#ipv6 route 3002:ABCE::/32 2002:AC10:101::
R4(config)#ipv6 route 2002:AC10:102::/48 Tunnel0
R4(config)#ipv6 route 3001::/16 2002:AC10:102::
```

d) Příkazmi **show interfaces tunnel0** a **show ipv6 interfaces tunnel0** si zobrazíme bližší informace o vytvořeném tunely na směrovači R3.

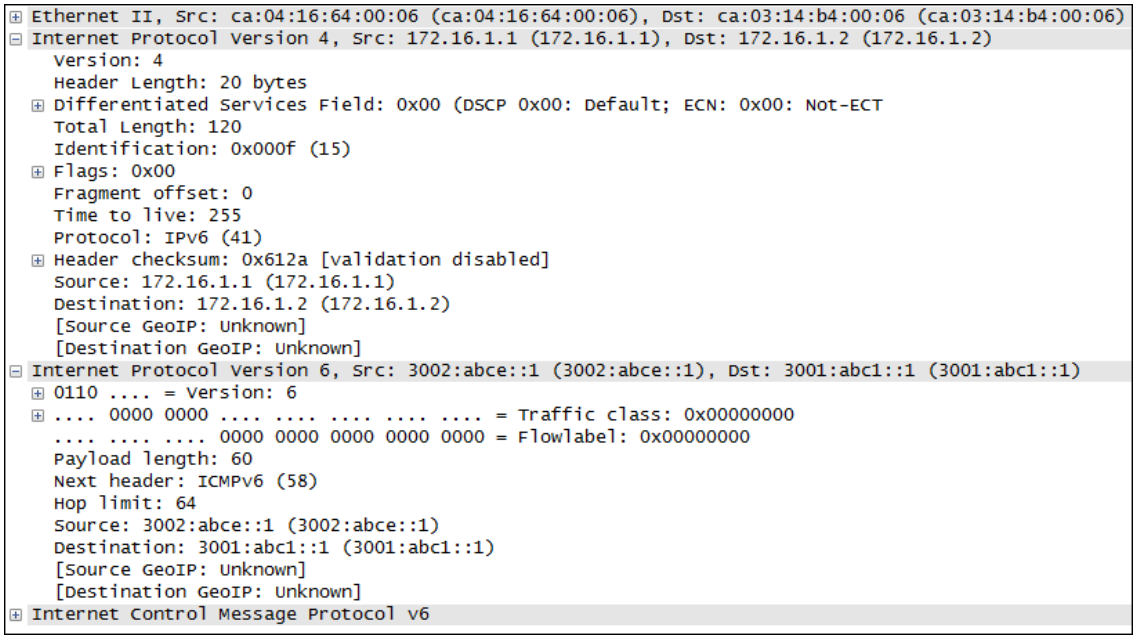
```
R3#show interfaces tunnel 0
Tunnel0 is up, line protocol is up
  Hardware is Tunnel
  MTU 17920 bytes, BW 100 Kbit/sec, DLY 50000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation TUNNEL, loopback not set
  Keepalive not set
  Tunnel source 172.16.1.2
  Tunnel protocol/transport IPv6 6to4
  Tunnel TTL 255
  Tunnel transport MTU 1480 bytes
<výstup zkrácen>
```

```
R3#show ipv6 interface tunnel 0
Tunnel0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::AC10:102
  No Virtual link-local address(es):
  Global unicast address(es):
    2002:AC10:102::, subnet is 2002:AC10:102::/48
<výstup zkrácen>
  MTU is 1480 bytes
<výstup zkrácen>
```

e) Na směrovači R4 rozhraní fastEthernet0/1 si spustíme zachytávání provozu. Ze směrovače R4 se budeme dotazovat prostřednictvím příkazu **ping** na cílovou adresu 3001:ABC1::1, přičemž zdrojová adresa bude loopback1.

```
R4#ping 3001:abc1::1 source loopback1
```

V zachyceném provozu si prohlédněte první ping dotaz. Obrázek A.10 ilustruje zapouzdření ICMP paketu v rámci tunelu. V hlavičce IPv4 protokolu je informace v poli *Protocol* s číslem 41, která indikuje, že paket v sobě nese IPv6 protokol. Hlavička IPv6 protokolu v poli *Next header* nese informaci s číslem 58 o použitém protokolu ICMP.



Obr. A.10: Zapouzdření ICMP paketu v rámci přechodu 6to4 tunelem.

A.1.10 Kontrolní otázky

1. Na základě čeho si směrovač stanoví *Router ID* a jak postupuje?
2. Jak se nazývají a jakou mají úlohu prefixy, nichž první kvartál je FE80.
3. Jaký je rozdíl mezi autentizací zpráv OSPFv2 a OSPFv3?
4. Proč je velikost MTU GRE tunelu jen 1476 bajtů a ne 1500 bajtů?
5. Proč je velikost MTU 6to4 tunelu jen 1480 bajtů a ne 1500 bajtů?
6. Proč se v debug výpisu OSPFv3 paketu nenachází pole *aut* jako u OSPFv2?

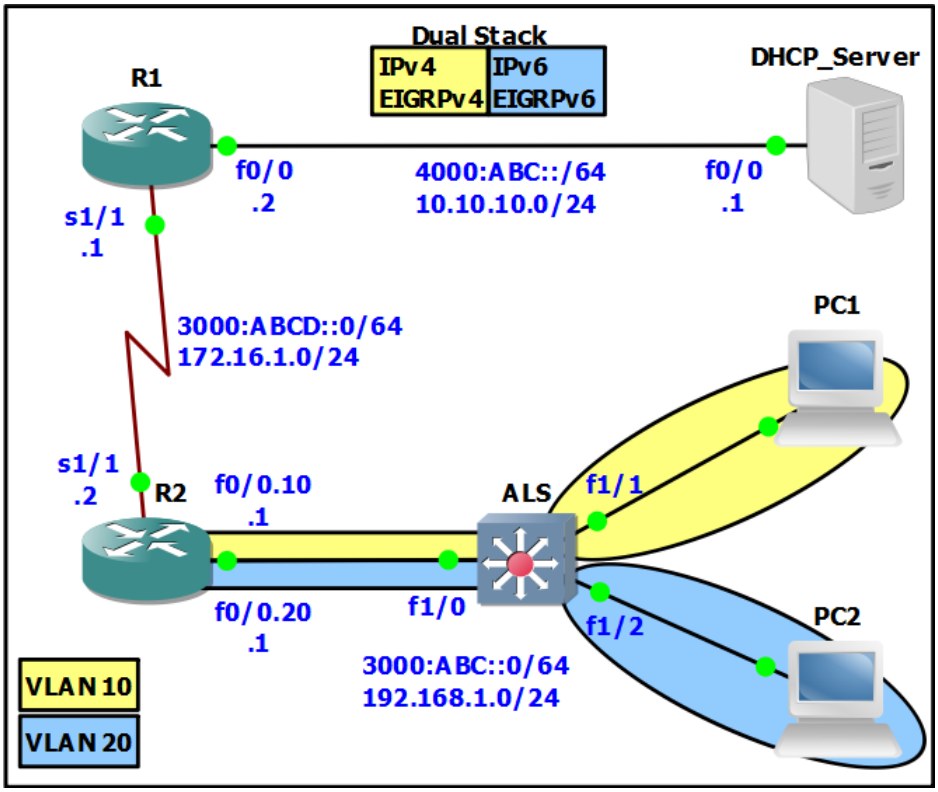
A.1.11 Samostatná úloha

Zrušte GRE tunel mezi směrovači R3 a R5. Upravte konfiguraci topologie tak, aby mezi směrovači R3, R4 a R5 byl nakonfigurován pouze tunel 6to4.

Poznámka: Přiložené DVD obsahuje konfiguraci zařízení před a po samostatné úloze. Dále obsahuje samotné zdrojové soubory projektů pro GNS3.

A.2 Laboratorní úloha 2

A.2.1 Zapojení topologie sítě



Obr. A.11: Zapojení topologie sítě.

A.2.2 Hlavní cíle úlohy

- Konfigurace směrovacího protokolu EIGRP
- Konfigurace směrovacího protokolu EIGRPv6
- Konfigurace DHCP a DHCPv6 serveru
- Rozbor ICMP protokolu pro IPv4 a IPv6
- Rozbor fragmentace v IPv4 a IPv6

A.2.3 Teoretický úvod

Směrovací protokol EIGRP se řadí mezi protokoly typu IGP a jeho aplikace je vhodná pro sítě různé velikosti [30]. Jedná se o proprietární protokol společnosti Cisco. Aplikační protokol DHCP je hojně využívaným protokolem v rámci síťové infrastruktury. Slouží pro automatickou konfiguraci adresních parametrů koncových

stanic. V rámci protokolového standardu IPv4 mluvíme o stavové DHCP konfigurací zatímco při verzí IPv6 se rozděluje DHCP na dva typy, stavový a bezestavový. V praxi se používá bezestavový DHCPv6. V tomto případě se adresa nezískává od serveru ale odvozuje se na základě získaných informací od lokálního směrovače. Protokol ICMP je často využívaným protokolem v rámci IP komunikace. Konkrétně u protokolu IPv6 je dokonce nedílnou součástí bez níž by nebyl funkční. K fragmentaci paketů dochází v případě pokud paket určité velikosti převyšuje velikost maximální přenášené jednotky (MTU). Protokoly IPv4 a IPv6 se liší ve způsobu fragmentace dat v rámci komunikace koncových uzlů. Zatímco u IPv4 docházelo k fragmentaci paketů i v rámci síťové infrastruktury u IPv6 se tento problém řeší zcela jiným způsobem. V tomto případě si samotný zdrojový odesílatel paketu zjistí maximální velikost přenášené jednotky, kterou daná síťová infrastruktura podporuje. Tuto informaci získá na základě ICMPv6 zpráv.

A.2.4 Úkol 1 – Konfigurace instance EIGRP

a) Jako první si nakonfigurujeme IP adresaci na jednotlivých rozhraních. Podle topologie na obrázku A.11 nakonfigurujte IPv4 adresy na všech rozhraních. Protokolem IPv6 se budeme zabývat později. Pro směrovač R2 využijeme konfiguraci sub–rozhraní fastEthernet0/0. Rozhraní nakonfigurujte následovně.

```
R2(config)#interface fastEthernet0/0.10
R2(config-subif)#encapsulation dot1Q 10
R2(config-subif)#ip address 192.168.1.1 255.255.255.0
```

b) Nyní nakonfigurujeme směrovací protokol EIGRP. Spustíme instanci EIGRP s ID autonomního systému 1 následujícím příkazem:

```
R1(config)#router eigrp 1
```

Nyní došlo k přidělení 32 bitového identifikátoru *Router ID*. Příkazem uvedeným níže si ověříme jeho hodnotu.

```
R1#show ip eigrp 1 topology
EIGRP-IPv4 Topology Table for AS(1)/ID(172.16.1.1)
<výstup zkrácen>
```

Nastavení směrovacího procesu proveďte i na směrovači R2 a DHCP serveru.

c) Jako další bod je přidání sítí do směrovacího procesu. V konfiguračním režimu směrovacího procesu přidáme přímo připojené sítě.

```
R1(config-router)#network adresa_sítě wildcard_maska
```

Obdobně nakonfigurujte směrovač R2 a DHCP server.

d) Po tomto kroku by měla mít síť IPv4 plnou konektivitu. Příkazem **ping** otestujte dostupnost sítě.

A.2.5 Úkol 2 – Konfigurace instance EIGRPv6

a) V tomto kroku si nakonfigurujeme základní IP adresaci sítě. Konfigurace IPv6 adresy na rozhraní se konfiguruje následovně:

```
R1(config-if)#ipv6 address adresa_sítě/prefix
```

I v tomto případě nakonfigurujeme na směrovači R2 sub–rozhraní. Rozhraní nakonfigurujte s indexem 20 následovně:

```
R2(config)#interface fastEthernet0/0.20
R2(config-subif)#encapsulation dot1Q 20
R2(config-subif)#ipv6 address 3000:ABC::1/64
```

b) Po tom, co jsou adresy nakonfigurovány, můžeme přistoupit ke konfiguraci směrovacího protokolu EIGRPv6. Jako první je třeba aktivovat směrování IPv6 příkazem:

```
R1(config)#ipv6 unicast-routing
```

Instance EIGRPv6 se konfiguruje obdobně jako to bylo u předchozí verze.

```
R1(config)#ipv6 router EIGRP číslo_instance
```

Implicitně je proces EIGRPv6 vypnutý. Proces je nutné spustit příkazem **no shutdown**. Následující příkazy aplikujte na směrovače R1, R2 a DHCP server.

```
R1(config)#ipv6 router EIGRP 1
R1(config-rtr)#no shutdown
```

c) Dalším krokem je přidání sítí do směrovacího procesu. V rámci IPv6 směrovacích protokolů je zažitý postup přidání sítí v konfigurací rozhraní. V tomto případě bude konfigurace následovná:

```
R1(config)#interface fastEthernet0/0 (resp. 0/0.20)
R1(config-if)#ipv6 eigrp číslo_instance
```

Tento krok proveďte na směrovačích R1, R2 a DHCP serveru.

d) S příkazy **show ipv6 eigrp topology** a **show ipv6 route** zkontrolujte topologii a směrovací tabulky směrovačů R1, R2 a DHCP serveru. Nyní by oblast IPv6 měla mít plnou konektivitu. Pro testování použijte příkaz **ping**.

e) Spusťte si zachytávání provozu na směrovači R1 rozhraní fastEthernet0/0. Obrázek A.12 ilustruje zachycené hello pakety. Všimněte si adresy, na které jsou tyto pakety zasílány. Pro verzi IPv4 je skupinová adresa typická prvním oktetem s dekadickou hodnotou 224. Protokol IPv6 používá hexadecimální hodnotu FF02. Poslední hodnota u obou zápisů představuje stejné číslo.

Source	Destination	Protocol	Length	Info
10.10.10.2	224.0.0.10	EIGRP	74	Hello
fe80::c802:17ff:fe9c:8	ff02::a	EIGRP	94	Hello
fe80::c804:10ff:fec4:8	ff02::a	EIGRP	94	Hello
10.10.10.1	224.0.0.10	EIGRP	74	Hello
fe80::c804:10ff:fec4:8	ff02::a	EIGRP	94	Hello
10.10.10.2	224.0.0.10	EIGRP	74	Hello
fe80::c802:17ff:fe9c:8	ff02::a	EIGRP	94	Hello

Obr. A.12: Zasílaní hello zpráv na směrovači R1.

A.2.6 Úkol 3 – Konfigurace přepínače ALS

a) Na novodobém Cisco přepínači se vytvářejí VLAN (Virtual LAN) sítě prostřednictvím následujících příkazů.

```
ALS(config)#vlan číslo_VLAN
ALS(config-vlan)#name meno_VLAN
```

V simulačním prostředí GNS3 nemáme tuto možnost, protože nepracujeme přímo s reálným přepínačem. Používáme směrovač, který je osazen EtherSwitch modulem a tím jsme schopni do jisté míry simulovat přepínač. EtherSwitch modul nedisponuje všemi vlastnostmi reálného přepínače. Jednou z nich je i nemožnost vytvořit VLAN klasickým výše uvedeným způsobem. V případě GNS3 se využívá starší způsob vytváření VLAN. Následující příkazy zobrazují tento způsob.

```
ALS#vlan database
ALS(vlan)#vlan 10 name IPV4
VLAN 10 modified:
    Name: IPV4
ALS(vlan)#vlan 20 name IPV6
VLAN 20 modified:
    Name: IPV6
ALS(vlan)#exit
```

Po zadání příkazu **exit** dojde k aplikování výše uvedených příkazů a tím k vytvoření požadovaných VLAN.

b) Klasickým způsobem, jako to známe s reálného přepínače, přidáme VLAN na konkrétní porty a ty nastavíme do příslušného pracovního režimu.

```
ALS(config)#interface fastEthernet 1/1
ALS(config-if)#switchport mode access
ALS(config-if)#switchport access vlan 10

ALS(config)#interface fastEthernet 1/2
ALS(config-if)#switchport mode access
ALS(config-if)#switchport access vlan 20

ALS(config)#interface fastEthernet 1/0
ALS(config-if)#switchport mode trunk
ALS(config-if)#switchport trunk allowed vlan all
```

c) Pokud chceme ověřit vytvořené VLAN a jejich přiřazení k rozhraním, použijeme následující příkaz, který je platný právě pro EtherSwitch modul.

```
ALS-SW#show vlan-switch
VLAN Name                 Status    Ports
---- -------------------- --------- -------------------------------
1    default              active    Fa1/3, Fa1/4, Fa1/5, Fa1/6
                                    Fa1/7, Fa1/8, Fa1/9, Fa1/10
                                    Fa1/11, Fa1/12, Fa1/13, Fa1/14
10   IPv4                 active    Fa1/1
20   IPv6                 active    Fa1/2
<výstup zkrácen>
```

A.2.7 Úkol 4 – Konfigurace DHCP Serveru

a) Nejprve si nakonfigurujeme službu DHCPv4. Na Cisco směrovačích se spouští DHCP služba za pomoci příkazu:

```
DHCP_Server(config)#service dhcp
```

Poznámka: Tato služba je implicitně spuštěna a není třeba ji manuálně spouštět. Informace je uvedena z důvodu, že služba může být z bezpečnostních důvodů explicitně zastavena.
V konfiguračním režimu vytvoříme DHCP pool s názvem IPV4 následovně:

```
DHCP_Server(config)#ip dhcp pool IPV4
```

b) Dále si nakonfigurujeme parametry jako adresu sítě, výchozí bránu, DNS server, doménové jméno a dobu zapůjčení adresních parametrů.

```
DHCP_Server(dhcp-config)#network 192.168.1.0 255.255.255.0
DHCP_Server(dhcp-config)#default-router 192.168.1.1
DHCP_Server(dhcp-config)#dns-server 70.70.70.70 8.8.8.8
DHCP_Server(dhcp-config)#domain-name example_IPv4.com
DHCP_Server(dhcp-config)#lease 1 2 30
```

Kde hodnota *lease* 1 2 30 reprezentuje dobu zapůjčení 1 den, 2 hodiny a 30 minut. Dále si všimněte způsob konfigurace DNS serverů. Tímto způsobem můžeme nakonfigurovat až 8 DNS serverů.

c) Dále je je nutné rezervovat jistý rozsah adres, které jsou většinou manuálně přiřazeny ke koncovým zařízením jako servery a tiskárny a samotné výchozí brány, jejichž adresy jsou statické. Na toto použijeme příkaz:

```
DHCP_Server(dhcp-config)#ip dhcp excluded-address 192.168.1.1
```

V rámci tohoto příkazu lze rezervovat jednu IP adresu nebo celý rozsah adres.

d) Jelikož samotné koncové stanice resp. PC1 se nenachází ve stejné síti jako je DHCP server, je třeba nakonfigurovat tzv. relay agenta. Relay agenta nakonfigurujeme na směrovači R2 příkazem:

```
R2(config)#interface fastEthernet0/0.10
R2(config-if)#ip helper-address A.B.C.D
```

Kde A.B.C.D je IP adresa DHCP serveru.

e) Nyní by měla být konfigurace DHCP korektně nakonfigurována. Spusťte zachytávání provozu na směrovači R2 rozhraní fastEthernet0/0. V rámci virtualizované koncové stanici PC1 si otevřeme příkazový řádek a s následujícími příkazy si ověříme funkčnost naší konfigurace:

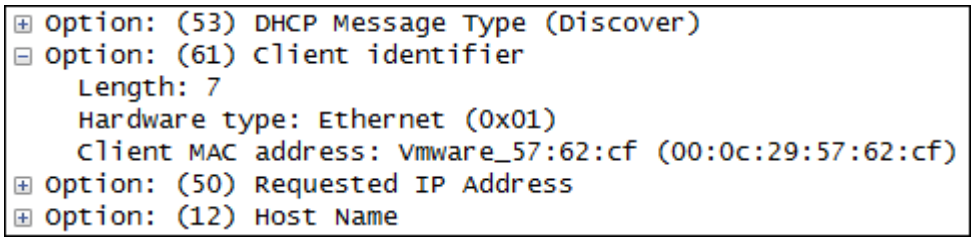
```
ipconfig /release
ipconfig /renew
```

Obrázek A.13 zachycuje proces získávání adresních parametrů. Samostatně prostudujte jednotlivá pole hlavičky DHCP protokolu pro jednotlivé zprávy.

Source	Destination	Protocol	Length	Info
0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover
192.168.1.1	192.168.1.2	DHCP	342	DHCP Offer
0.0.0.0	255.255.255.255	DHCP	356	DHCP Request
192.168.1.1	192.168.1.2	DHCP	342	DHCP ACK
192.168.1.2	255.255.255.255	DHCP	342	DHCP Inform
192.168.1.1	255.255.255.255	DHCP	342	DHCP ACK

Obr. A.13: Zachycený provoz na směrovači R2 po příkazu renew.

f) Rozklikněte si zprávu DHCP Discover. Obrázek A.14 ilustruje identifikaci konkrétní koncové stanice v rámci DHCP procesu. Jak je vidět z obrázku, identifikace je založena na základě MAC adresy.



Obr. A.14: Obsah pole Option – Client identifier protokolu Bootstrap.

g) Na DHCP servery si příslušným příkazem zobrazíme záznamy poskytnutých adresních parametrů pro jednotlivé koncové zařízení.

```
DHCP_Server#show ip dhcp binding
Bindings from all pools not associated with VRF:
IP address   Client-ID          Lease expiration      Type
             Hardware address
             User name
192.168.1.2  0100.0c29.5762.cf  Jan 25 2015 05:48 PM  Automatic
```

Z výše uvedeného výpisu je patrné, že DHCP server si vede záznamy vzhledem k MAC adresám koncových stanic.

A.2.8 Úkol 5 – Konfigurace stavového DHCPv6 Serveru

a) V konfiguračním režimu nejdříve vytvoříme DHCP pool s názvem IPV6. Tímto se zároveň dostaneme do režimu konfigurace DHCPv6.

```
DHCP_Server(config)#ipv6 dhcp pool IPV6
```

Nakonfigurujeme prefix, ze kterého bude DHCPv6 přidělovat adresy koncovým stanicím.

```
DHCP_Server(config-dhcpv6)#address prefix 3000:ABC::/64
lifetime 3600 1800
```

Kde údaj 3600 charakterizuje hodnotu *valid time* a údaj 1800 charakterizuje hodnotu *preferred time*. Časy jsou udány v sekundách. Hodnota *valid time* slouží pro server a klienta zároveň. Po uplynutí této doby klient ztrácí přidělené adresní nastavení. Hodnota *preferred time* slouží čistě jen pro klienta. Podmínkou je, aby *valid time* byl větší nebo minimálně stejný jako preferred time.

b) Dále si nastavíme adresy DNS serverů a doménového jména.

```
DHCP_Server(config-dhcpv6)#dns-server AAAA:BBBB::1
DHCP_Server(config-dhcpv6)#dns-server CCCC:DDDD::1
DHCP_Server(config-dhcpv6)#domain-name IPv6_example.com
```

Všimněte si rozdílu konfigurace DNS serverů. Zatímco u IPv4 verze se nakonfigurovali DNS servery v rámci jednoho příkazu, ve verzi IPv6 se toto řeší odděleně.

c) Dalším krokem je aktivování DHCPv6 serveru. Toto se provede příkazem v rámci konfiguračního režimu rozhraní.

```
DHCP_Server(config)#interface fastEthernet0/0
DHCP_Server(config-if)#ipv6 dhcp server IPV6
```

Kde IPV6 reprezentuje náš DHCPv6 pool.

d) V rámci toho, že koncové stanice resp. PC2 se nachází mimo síť DHCPv6 serveru, je nutné nakonfigurovat na směrovači R2 relay agenta. Toto provedeme příkazem:

```
R2(config)#interface fastEthernet0/0.20
R2(config-subif)#ipv6 dhcp relay destination 4000:ABC::1
```

e) V rámci DHCP pro IPv6 je nutné nastavit na směrovači R2 tzv. NDP parametry. Konfigurace se provádí na sub–rozhraní fastEthernet0/0.20

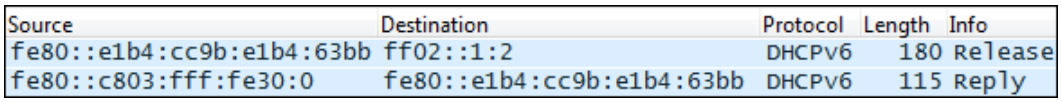
```
R2(config-subif)#ipv6 nd other-config-flag
R2(config-subif)#ipv6 nd managed-config-flag
```

Parametr *other–config–flag* slouží k získání parametrů jako DNS adresy, zatímco *managed–config–flag* slouží k získání parametrů jako je samotná IPv6 adresa. Pokud zakážeme tento bit mluvíme o bezestavové konfiguraci. V případě povolení obou parametrů mluvíme o stavovém DHCPv6.

f) Nyní by měla být konfigurace DHCPv6 korektně nakonfigurována. Spusťte zachytávání provozu na směrovači R2 rozhraní fastEthernet0/0. V rámci virtualizované koncové stanici PC2 si otevřeme příkazový řádek a s následujícími příkazy si ověříme funkčnost naší konfigurace:

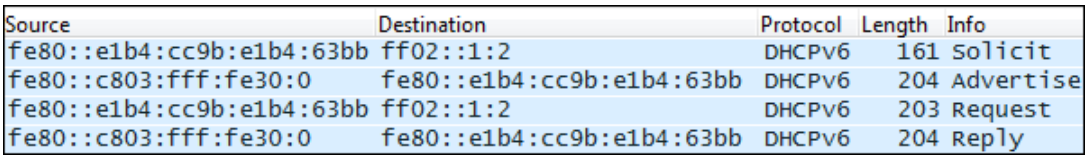
```
ipconfig /release6
ipconfig /renew6
```

Obrázek A.15 a A.16 zachycuje proces získávání adresních parametrů. Samostatně prostudujte jednotlivá pole hlavičky protokolu DHCPv6.



Source	Destination	Protocol	Length	Info
fe80::e1b4:cc9b:e1b4:63bb	ff02::1:2	DHCPv6	180	Release
fe80::c803:fff:fe30:0	fe80::e1b4:cc9b:e1b4:63bb	DHCPv6	115	Reply

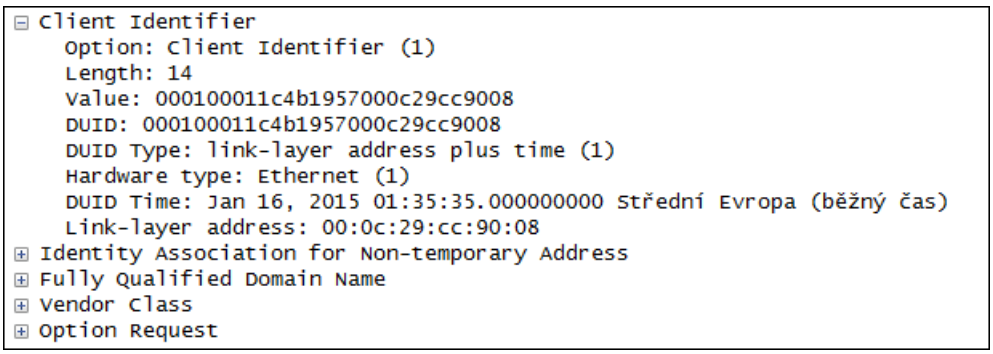
Obr. A.15: Zachycený provoz na směrovači R2 po příkazu release.



Source	Destination	Protocol	Length	Info
fe80::e1b4:cc9b:e1b4:63bb	ff02::1:2	DHCPv6	161	Solicit
fe80::c803:fff:fe30:0	fe80::e1b4:cc9b:e1b4:63bb	DHCPv6	204	Advertise
fe80::e1b4:cc9b:e1b4:63bb	ff02::1:2	DHCPv6	203	Request
fe80::c803:fff:fe30:0	fe80::e1b4:cc9b:e1b4:63bb	DHCPv6	204	Reply

Obr. A.16: Zachycený provoz na směrovači R2 po příkazu renew.

g) Rozklikněte si zprávu DHCP Solicit. Obrázek A.17 ilustruje identifikaci konkrétní koncové stanice v rámci DHCP procesu. Jak je vidět z obrázku, identifikace je založena na základě DUID parametrů.



```
⊟ Client Identifier
    Option: Client Identifier (1)
    Length: 14
    Value: 000100011c4b1957000c29cc9008
    DUID: 000100011c4b1957000c29cc9008
    DUID Type: link-layer address plus time (1)
    Hardware type: Ethernet (1)
    DUID Time: Jan 16, 2015 01:35:35.000000000 Střední Evropa (běžný čas)
    Link-layer address: 00:0c:29:cc:90:08
⊞ Identity Association for Non-temporary Address
⊞ Fully Qualified Domain Name
⊞ Vendor Class
⊞ Option Request
```

Obr. A.17: Obsah pole Client identifier protokolu DHCPv6.

h) Na DHCPv6 servery si příslušným příkazem zobrazíme záznamy poskytnutých adresních parametrů pro jednotlivé koncové zařízení.

```
DHCP_Server#show ipv6 dhcp binding
Client: FE80::E1B4:CC9B:E1B4:63BB
  DUID: 000100011C4B1957000C29CC9008
  Username : unassigned
  IA NA: IA ID 0x11000C29, T1 900, T2 1440
    Address: 3000:ABC::A897:B1DE:8713:DD4
             preferred lifetime 1800, valid lifetime 3600
             expires at Jan 25 2015 07:43 PM (3453 seconds)
```

V tomto případě si DHCPv6 server vede záznamy poskytnutých adresních parametrů v rámci hodnoty DUID klienta. V rámci virtualizované koncové stanici PC2 si příkazem **ipconfig /all** zobrazíme adresní parametry rozhraní.

Obrázek A.18 poukazuje na fakt, že adresa je zapůjčena. Třeba ale podotknout, že tato forma DHCPv6 se standardně nepoužívá. V běžném provozu se setkáme s bezstavovým DHCPv6.

```
C:\Windows\System32>ipconfig /all
Adaptér sítě Ethernet Připojení k místní síti:

   Přípona DNS podle připojení . . . : IPv6_example.com
   Popis . . . . . . . . . . . . . . : Intel(R) PRO/1000 MT - síťové připojení
   Fyzická Adresa. . . . . . . . . . : 00-0C-29-CC-90-08
   Protokol DHCP povolen . . . . . . : Ano
   Automatická konfigurace povolena  : Ano
   IPv6 adresa. . . . . . . . . . . . : 3000:abc::a897:b1de:8713:dd4(Preferované)

   Zapůjčeno . . . . . . . . . . . . : 25. ledna 2015 19:00:20
   Zápůjčka vyprší . . . . . . . . . : 25. ledna 2015 20:00:20
   IPv6 adresa. . . . . . . . . . . . : 3000:abc::e1b4:cc9b:e1b4:63bb(Preferované)

   Dočasná IPv6 adresa. . . . . . . . : 3000:abc::20d0:8af1:b364:1cf9(Preferované)
   Místní IPv6 adresa v rámci propojení . . . : fe80::e1b4:cc9b:e1b4:63bb%14(Pre
ferované)
   Adresa IP automatické konfigurace : 169.254.99.187(Preferované)
   Maska podsítě . . . . . . . . . . : 255.255.0.0
   Výchozí brána . . . . . . . . . . : fe80::c803:fff:fe30:0%14
   IAID DHCPv6 . . . . . . . . . . . : 285215785
   DUID klienta DHCPv6. . . . . . . . : 00-01-00-01-1C-4B-19-57-00-0C-29-CC-90-08

   Servery DNS . . . . . . . . . . . : aaaa:bbbb::1
                                       cccc:dddd::1
   Rozhraní NetBios nad protokolem TCP/IP. . . . . . . . : Povoleno
   Seznam hledání přípon DNS specifických pro připojení:
                                       IPv6_example.com
```

Obr. A.18: Výpis adresních parametrů stanice PC2 (stavový DHCPv6).

A.2.9 Úkol 6 – Konfigurace bezstavového DHCPv6 Serveru

a) Nyní přistoupíme ke konfiguraci bezstavového DHCPv6, který je zároveň standardně využíván v rámci IPv6. Docílíme toho tím, že na směrovači R2 zadáme příkaz pro vypnutí zasílání tzv. manažmentových parametrů.

```
R2(config)#interface fastEthernet0/0.20
R2(config-subif)#no ipv6 nd managed-config-flag
```

b) Pro názornost si taky můžeme zrušit přidělování adres z definovaného rozsahu, které jsme nastavili na serveru DHCPv6.

```
DHCP_Server(config)#ipv6 dhcp pool IPV6
DHCP_Server(config-dhcpv6)#no address prefix 3000:ABC::/64
lifetime 3600 1800
```

Takto si můžeme být naprosto jistý, že server určitě nebude přidělovat adresu pro koncová zařízení. V tomto případě slouží server čistě jen na přidělování parametrů jako jsou DNS servery a doménové jméno.

c) Provedeme restart rozhraní PC2, které je připojeno do naší simulované sítě. K tomuto využijeme příkazy pro příkazový řádek systému Windows.

```
netsh interface set interface name="Připojení k místní síti"
admin=disabled

netsh interface set interface name="Připojení k místní síti"
admin=enabled
```

V rámci virtualizované koncové stanici PC2 si příkazem **ipconfig /all** zobrazíme adresní parametry rozhraní. Obrázek A.32 poukazuje na fakt, že adresa není zapůjčena.

```
C:\Windows\System32>ipconfig /all
Adaptér sítě Ethernet Připojení k místní síti:

   Přípona DNS podle připojení . . . : IPv6_example.com
   Popis . . . . . . . . . . . . . . : Intel(R) PRO/1000 MT - síťové připojení
   Fyzická Adresa. . . . . . . . . . : 00-0C-29-CC-90-08
   Protokol DHCP povolen . . . . . . : Ano
   Automatická konfigurace povolena  : Ano
   IPv6 adresa. . . . . . . . . . . . : 3000:abc::e1b4:cc9b:e1b4:63bb(Preferované)

   Dočasná IPv6 adresa. . . . . . . . : 3000:abc::bd59:7c1d:d8ee:3123(Preferované)
   Místní IPv6 adresa v rámci propojení . . . : fe80::e1b4:cc9b:e1b4:63bb%14(Pre
ferované)
   Výchozí brána . . . . . . . . . . . : fe80::c803:fff:fe30:0%14
   IAID DHCPv6 . . . . . . . . . . . . : 285215785
   DUID klienta DHCPv6. . . . . . . . : 00-01-00-01-1C-4B-19-57-00-0C-29-CC-90-08

   Servery DNS . . . . . . . . . . . : aaaa:bbbb::1
                                       cccc:dddd::1
   Rozhraní NetBios nad protokolem TCP/IP. . . . . . . . . : Povoleno
   Seznam hledání přípon DNS specifických pro připojení:
                                       IPv6_example.com
```

Obr. A.19: Výpis adresních parametrů stanice PC2 (bezstavový DHCPv6).

A.2.10 Úkol 7 – Protokol ICMPv6

Spusťte si zachytávání provozu na rozhraní Serial1/1 směrovače R2. Vypněte a následně zapněte rozhraní Serial1/1 směrovače R1 příkazem **(no)shutdown**. Vyfiltrujte si ICMPv6 zprávy. Měli byste být schopení vidět zprávy jako jsou ilustrovány na obrázku A.20.

Source	Destination	Protocol	Length	Info
3000:abcd::bd59:7c1d:{ff02::1:ff30:0		ICMPv6	90	Neighbor Solicitation
fe80::c803:fff:fe30:0	3000:abcd::bd59:7c1d:d8ee:3123	ICMPv6	90	Neighbor Advertisement

Obr. A.20: Zprávy typu ND (Neighbor Discovery).

Po rozkliknutí zprávy Neighbor Solicitation vidíme, že se jedná o ICMPv6 zprávu typu 135. Toto ilustruje obrázek A.21. Tyto zprávy slouží ke zjišťování linkové lokální adresy sousedního směrovače. Jakékoliv zařízení, které pracuje s IPv6, je schopno vyslat tuto zprávu. Cílová adresa je linková lokální adresa směrovače R2 rozhraní

fastEthernet0/0. Pole *Option* nese informaci o zdrojové fyzické adrese a tou je MAC adresa rozhraní PC2.

```
Internet Control Message Protocol v6
    Type: Neighbor Solicitation (135)
    Code: 0
    Checksum: 0xf1b0 [correct]
    Reserved: 00000000
    Target Address: fe80::c803:fff:fe30:8 (fe80::c803:fff:fe30:8)
  ICMPv6 Option (Source link-layer address : 02:00:4c:4f:4f:50)
      Type: Source link-layer address (1)
      Length: 1 (8 bytes)
      Link-layer address: 02:00:4c:4f:4f:50 (02:00:4c:4f:4f:50)
```

Obr. A.21: Zpráva typu NS (Neighbor Solicitation).

Obrázek A.22 ilustruje zprávu Neighbor Advertisement. Jedná se o zprávu s ICMPv6 číslem 136 a je odpovědí na zprávu typu Neighbor Solicitation. Informace o cílové adrese nese linkovou lokální adresu, na kterou byla zaslána původní zpráva Neighbor Solicitation. Položka *Option* nese informaci o MAC adrese, která reprezentuje rozhraní fastEthernet0/0 směrovače R2. Příznak *Router* s hodnotou 1 indikuje, že odesílatel NA zprávy je směrovač. Hodnota 0 by indikovala například koncovou stanici. Příznak *Solicited* indikuje, že se jedná o zprávu NA, která byla vyžádána zprávou NS. Příznak *Override* indikuje, že odpověď NA může nahradit stávající položky v dočasné paměti sousedství.

```
Internet Control Message Protocol v6
    Type: Neighbor Advertisement (136)
    Code: 0
    Checksum: 0xfd93 [correct]
  Flags: 0xe0000000
      1... .... .... .... .... .... .... .... = Router: Set
      .1.. .... .... .... .... .... .... .... = Solicited: Set
      ..1. .... .... .... .... .... .... .... = Override: Set
      ...0 0000 0000 0000 0000 0000 0000 0000 = Reserved: 0
    Target Address: fe80::c803:fff:fe30:8 (fe80::c803:fff:fe30:8)
  ICMPv6 Option (Target link-layer address : ca:03:0f:30:00:08)
      Type: Target link-layer address (2)
      Length: 1 (8 bytes)
      Link-layer address: ca:03:0f:30:00:08 (ca:03:0f:30:00:08)
```

Obr. A.22: Zpráva typu NA (Neighbor Advertisement).

Příslušným debug výpisem si zobrazíme ICMPv6 provoz na směrovači R2.

```
R2#debug ipv6 icmp
  ICMP Packet debugging is on
ICMPv6:Received N-Solicit, Src=FE80::E1B4:CC9B:E1B4:63BB, Dst=FF02::1:FF30:0
ICMPv6:Sent N-Advert, Src=FE80::C803:FFF:FE30:0, Dst=FE80::E1B4:CC9B:E1B4:63BB
```

A.2.11 Úkol 8 – Fragmentace v IPv4

a) Jako první, změníme hodnotu MTU linky mezi směrovači R1 a R2. Nakonfigurujeme rozhraní směrovačů R1 a R2 příkazy:

```
R1(config)#interface serial1/1
R1(config-if)#mtu 1280
```

b) Spustíme zachytávání provozu na směrovači R2 fastEthernet0/0. Po tom jak se nám otevře program Wireshark si nastavíme prostředí tak aby se nám zobrazovali fragmentované data. Z horní lišty nabídky si rozklikneme položku *Edit*. Otevřeme nabídku *Preferences* a rozklikeneme nabídku *Protocols*. V ní si najdeme položku *IPv4* a zakážeme znovu seskupení IPv4 datagramů (*Reasemble fragmented IPv4 datagrams*). Totéž uděláme pro položku IPv6.

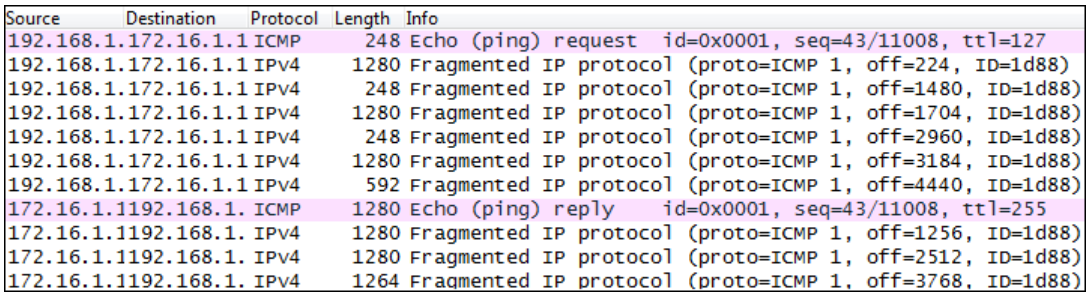
c) Příkazem **ping** se budeme dotazovat na cílovou adresu DHCP serveru, přičemž si nastavíme parametr pro rozšířený příkaz **ping**. Příznak –l určuje parametr pro velikost odesílaných ICMP dotazů v bajtech. Odesílat budeme ICMP dotazy velikosti 5000 bajtů.

```
ping 10.10.10.1 -l 5000
```

d) Na obrázku A.23 je zachycen provoz na směrovači R2 rozhraní fastEthernet0/0. Z obrázku je vidět, že jednotky dat, které přicházejí na rozhraní mají velikost 1518 bajtů. Je to s toho důvodu, že linka, která spojuje koncovou stanici a směrovač R2 má nastavenou hodnotu MTU velikosti 1500 bajtů. Zbylých 18 bajtů tvoří 14 bajtový Ethernet rámec a 4 bajtové CRC (Cyclic Redundancy Check). Na lince mezi R1 a R2 dojde k další fragmentaci, protože samotná 1518 bajtové jednotka je větší než námi nastavených 1280 bajtů, tak jak je to znázorněno na obrázku A.24.

Source	Destination	Protocol	Length	Info
192.168.1.2	10.10.10.1	ICMP	1518	Echo (ping) request id=0x0001, seq=5/1280, ttl=128
192.168.1.2	10.10.10.1	IPv4	1518	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=0a19)
192.168.1.2	10.10.10.1	IPv4	1518	Fragmented IP protocol (proto=ICMP 1, off=2960, ID=0a19)
192.168.1.2	10.10.10.1	IPv4	606	Fragmented IP protocol (proto=ICMP 1, off=4440, ID=0a19)
10.10.10.1	192.168.1.2	ICMP	262	Echo (ping) reply id=0x0001, seq=5/1280, ttl=253
10.10.10.1	192.168.1.2	IPv4	1294	Fragmented IP protocol (proto=ICMP 1, off=224, ID=0a19)
10.10.10.1	192.168.1.2	IPv4	262	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=0a19)
10.10.10.1	192.168.1.2	IPv4	1294	Fragmented IP protocol (proto=ICMP 1, off=1704, ID=0a19)
10.10.10.1	192.168.1.2	IPv4	262	Fragmented IP protocol (proto=ICMP 1, off=2960, ID=0a19)
10.10.10.1	192.168.1.2	IPv4	1294	Fragmented IP protocol (proto=ICMP 1, off=3184, ID=0a19)
10.10.10.1	192.168.1.2	IPv4	606	Fragmented IP protocol (proto=ICMP 1, off=4440, ID=0a19)

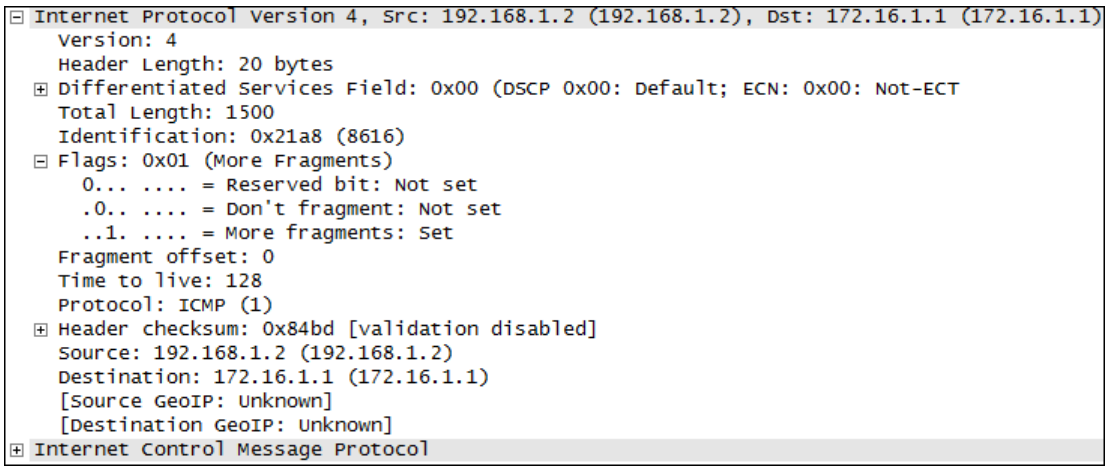
Obr. A.23: Fragmentace paketů mezi PC1 a R2.



Source	Destination	Protocol	Length	Info
192.168.1.1	72.16.1.1	ICMP	248	Echo (ping) request id=0x0001, seq=43/11008, ttl=127
192.168.1.1	72.16.1.1	IPv4	1280	Fragmented IP protocol (proto=ICMP 1, off=224, ID=1d88)
192.168.1.1	72.16.1.1	IPv4	248	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=1d88)
192.168.1.1	72.16.1.1	IPv4	1280	Fragmented IP protocol (proto=ICMP 1, off=1704, ID=1d88)
192.168.1.1	72.16.1.1	IPv4	248	Fragmented IP protocol (proto=ICMP 1, off=2960, ID=1d88)
192.168.1.1	72.16.1.1	IPv4	1280	Fragmented IP protocol (proto=ICMP 1, off=3184, ID=1d88)
192.168.1.1	72.16.1.1	IPv4	592	Fragmented IP protocol (proto=ICMP 1, off=4440, ID=1d88)
172.16.1.1	192.168.1.	ICMP	1280	Echo (ping) reply id=0x0001, seq=43/11008, ttl=255
172.16.1.1	192.168.1.	IPv4	1280	Fragmented IP protocol (proto=ICMP 1, off=1256, ID=1d88)
172.16.1.1	192.168.1.	IPv4	1280	Fragmented IP protocol (proto=ICMP 1, off=2512, ID=1d88)
172.16.1.1	192.168.1.	IPv4	1264	Fragmented IP protocol (proto=ICMP 1, off=3768, ID=1d88)

Obr. A.24: Fragmentace paketů mezi R1 a R2.

e) Zobrazíme si obsah hlavičky IP protokolu prvního paketu na dotaz příkazu **ping** ze zachyceného provozu na směrovači R2. Měli bychom vidět podobný výpis jako je na obrázku A.25. Z obrázku je vidět, že k danému segmentu bude přislouchat další část. Na toto poukazuje konkrétní příznak *More fragments*. Samostatně si zobrazte poslední část datové jednotky. Příznak *More fragments* by měl mít hodnotu 0.



```
⊟ Internet Protocol Version 4, Src: 192.168.1.2 (192.168.1.2), Dst: 172.16.1.1 (172.16.1.1)
    Version: 4
    Header Length: 20 bytes
  ⊞ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT
    Total Length: 1500
    Identification: 0x21a8 (8616)
  ⊟ Flags: 0x01 (More Fragments)
      0... .... = Reserved bit: Not set
      .0.. .... = Don't fragment: Not set
      ..1. .... = More fragments: Set
    Fragment offset: 0
    Time to live: 128
    Protocol: ICMP (1)
  ⊞ Header checksum: 0x84bd [validation disabled]
    Source: 192.168.1.2 (192.168.1.2)
    Destination: 172.16.1.1 (172.16.1.1)
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
⊞ Internet Control Message Protocol
```

Obr. A.25: Obsah hlavičky IP protokolu v závislosti na fragmentaci.

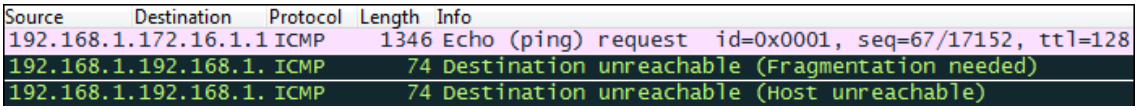
f) Na obrázku A.23 konkrétně dotazu ping vyšetřujte pole *Offset*. První část fragmentu by měla mít hodnotu 0. Druhá část bude mít 1480, třetí část 2960 a poslední část bude mít hodnotu 4440.

g) Na stanici PC1 s příkazového řádku zadejte následující příkaz.

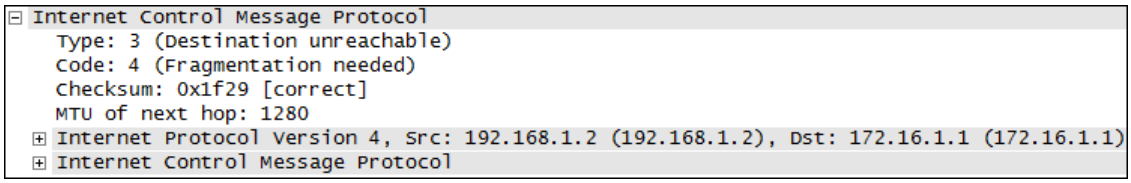
```
ping 10.10.10.1 -l 1300 -f
```

Příznak –f nastaví v IP hlavičce pole DF na hodnotu 1. Hodnota MTU 1300 je dostatečně malá aby směrovač R2 přijal dotaz, ale zároveň je příliš velká na to, aby byl tento dotaz přeposlán na směrovač R1. Zobrazte si zachycený provoz na směrovači R2, měli byste vidět výpis podobný jako ilustruje obrázek

A.26. Po rozkliknutí druhého paketu ICMP s popisem *Fragmentation needed*, obrázek A.27, je vidět, že taková ICMP zpráva je typu 3 a nese kód 4.



Obr. A.26: Zachycený provoz na směrovači R2 při nastaveném poli DF.



Obr. A.27: Hlavička ICMP paketu pri ohlašování potřeby fragmentace.

A.2.12 Úkol 9 – Fragmentace v IPv6

a) Na směrovači R2 fastEthernet0/0 spustíme zachytávání provozu a zároveň spustíme debug výpisi o ICMPv6 zprávách příkazem:

```
R2#debug ipv6 icmp
```

b) Na virtualizované stanici PC2 v příkazovém řádku provedeme smazání dočasných záznamů o MTU pro cílové adresy. Docílíme to příkazem:

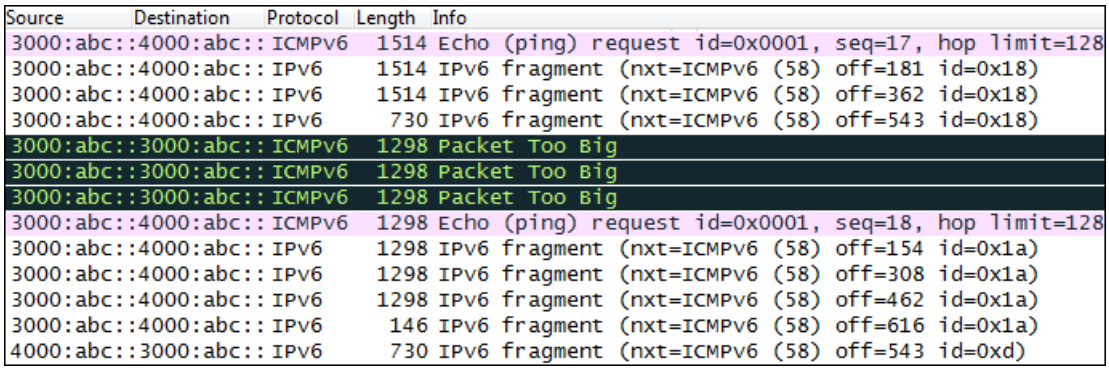
```
netsh interface ipv6 delete destinationcache
```

c) Na stanici PC2 s příkazového řádku zadejte příkaz:

```
ping 4000:abc::1 -l 5000
```

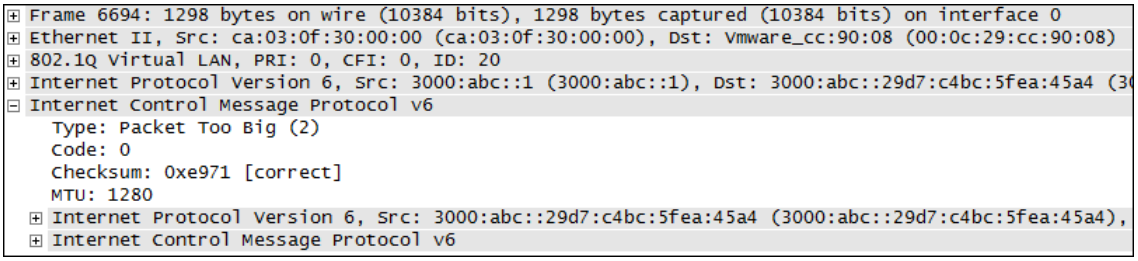
d) Měli byste být schopni zachytit provoz jako je znázorněn na obrázku A.28. Koncová stanice na začátku vyšle fragmentované pakety, protože v dočasné paměti má MTU údaje o svém rozhraní. Směrovač R2 však nedokáže přeposlat 5000 bajtové pakety přes sériové rozhraní. ICMPv6 zprávou typu *Packet Too Big* oznámí koncové stanici maximální velikost, kterou daná linka podporuje.

e) Po rozkliknutí zprávy *Packet Too Big*, obrázek A.29, vidíme další podrobnosti jako typ zprávy s číslem 2 a informaci o MTU a to je 1280 bajtů.



Source	Destination	Protocol	Length	Info
3000:abc::4000:abc::		ICMPv6	1514	Echo (ping) request id=0x0001, seq=17, hop limit=128
3000:abc::4000:abc::		IPv6	1514	IPv6 fragment (nxt=ICMPv6 (58) off=181 id=0x18)
3000:abc::4000:abc::		IPv6	1514	IPv6 fragment (nxt=ICMPv6 (58) off=362 id=0x18)
3000:abc::4000:abc::		IPv6	730	IPv6 fragment (nxt=ICMPv6 (58) off=543 id=0x18)
3000:abc::3000:abc::		ICMPv6	1298	Packet Too Big
3000:abc::3000:abc::		ICMPv6	1298	Packet Too Big
3000:abc::3000:abc::		ICMPv6	1298	Packet Too Big
3000:abc::4000:abc::		ICMPv6	1298	Echo (ping) request id=0x0001, seq=18, hop limit=128
3000:abc::4000:abc::		IPv6	1298	IPv6 fragment (nxt=ICMPv6 (58) off=154 id=0x1a)
3000:abc::4000:abc::		IPv6	1298	IPv6 fragment (nxt=ICMPv6 (58) off=308 id=0x1a)
3000:abc::4000:abc::		IPv6	1298	IPv6 fragment (nxt=ICMPv6 (58) off=462 id=0x1a)
3000:abc::4000:abc::		IPv6	146	IPv6 fragment (nxt=ICMPv6 (58) off=616 id=0x1a)
4000:abc::3000:abc::		IPv6	730	IPv6 fragment (nxt=ICMPv6 (58) off=543 id=0xd)

Obr. A.28: Fragmentace paketů mezi PC2 a DHCP serverem.



Obr. A.29: Obsah hlavičky ICMPv6 typu Packet Too Big.

Zároveň v debug výpisu obdržíme následující informace.

```
ICMP Packet debugging is on
ICMPv6: Sent Too Big about 4000:ABC::1, MTU=1280, Src=3000:ABC::1,
Dst=3000:ABC::20D0:8AF1:B364:1CF9
ICMPv6: Sent Too Big about 4000:ABC::1, MTU=1280, Src=3000:ABC::1,
Dst=3000:ABC::20D0:8AF1:B364:1CF9
ICMPv6: Sent Too Big about 4000:ABC::1, MTU=1280, Src=3000:ABC::1,
Dst=3000:ABC::20D0:8AF1:B364:1CF9
```

f) Na koncové stanici PC2 si zobrazíme dočasné údaje o MTU parametrech, které se koncová stanice naučila. Použijeme na to příkaz:

`netsh interface ipv6 show destinationcache`
`interface="`*`Připojení k místní síti`*`"`

Měli byste obdržet výpis jako ilustruje obrázek A.30. Z obrázku je vidět, že cílová adresa 4000:abc::1 je dostupná přes síť, jejíž MTU je 1280 bajtů.

```
C:\Windows\System32>netsh interface ipv6 show destinationcache interface="Připoj
ení k místní síti"

Rozhraní 14: Připojení k místní síti


PMTU Cílová adresa                                Adresa dalšího směrování
---- ------------------------------------------- ------------------------
1500 3000:abc::b819:c5b:d79f:8670                 3000:abc::b819:c5b:d79f:8670
1500 3000:abc::e1b4:cc9b:e1b4:63bb                3000:abc::e1b4:cc9b:e1b4:63bb

1280 4000:abc::1                                  fe80::c803:fff:fe30:0
1500 aaaa:bbbb::1                                 fe80::c803:fff:fe30:0
1500 fe80::e1b4:cc9b:e1b4:63bb                    fe80::e1b4:cc9b:e1b4:63bb
1500 3000:abc::20d0:8af1:b364:1cf9                3000:abc::20d0:8af1:b364:1cf9
```

Obr. A.30: Tabulka parametrů MTU na koncové stanici PC2.

A.2.13 Kontrolní otázky

1. Při chybném zápisu IPv4 adresy na rozhraní směrovače stačí přepsat adresu příkazem **ip address**. V případě IPv6 to neplatí. Proč?
2. Jakou výhodu má přidání sítě do směrovacího procesu v konfiguračním režimu rozhraní oproti zadání pomocí příkazu network?
3. Čím je způsobeno to, že rámec EIGRPv6 je o 20 bajtů větší než rámec EIGRP?
4. Který parametr z DHCPv4 je ekvivalentem příkazu **valid time** v DHCPv6?
5. V čem spočívá nevýhoda identifikace koncové stanice na základě MAC adresy u DHCP protokolu?
6. Jaký příkaz aktivuje rozesílání ICMPv6 zpráv?
7. V čem spočívá výhoda IPv6 s ohledem na řešení fragmentace dat?

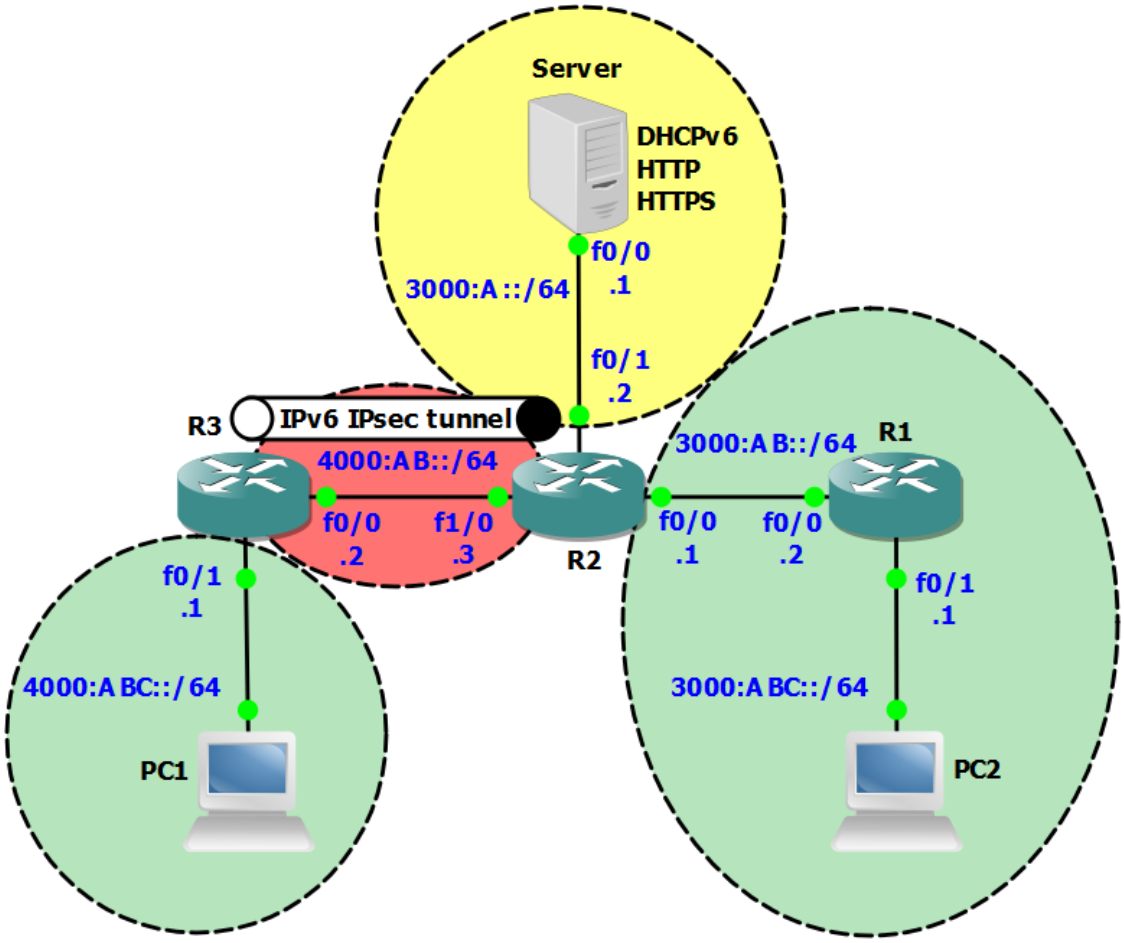
A.2.14 Samostatná úloha

Nakonfigurujte DHCPv4 a DHCPv6 servery tak aby přidělovali adresní parametry pouze na časový interval jedné minuty. Sledujte pomocí Wiresharku provoz a zjistěte kdy si koncové stanice požádají o prodloužení adresních parametrů.

Poznámka: Přiložené DVD obsahuje konfiguraci zařízení před a po samostatné úloze. Dále obsahuje samotné zdrojové soubory projektů pro GNS3. Přiložené DVD neobsahuje virtualizované stanice v programu VMware. V případě další práce s laboratorními úlohami je možné využít i VirtualBox. V rámci virtualizovaných stanic Windows 7 není nutná žádná specifická konfigurace.

A.3 Laboratorní úloha 3

A.3.1 Zapojení topologie sítě



Obr. A.31: Zapojení topologie sítě.

A.3.2 Hlavní cíle úlohy

- Konfigurace OSPFv3.
- Konfigurace DHCP serveru.
- Zabezpečení OSPFv3.
- Konfigurace IPv6 IPsec tunelu.
- Konfigurace IPv6 přístupových seznamů.
- Konfigurace IPv6 IOS firewallu.

A.3.3 Teoretický úvod

Význam autentizace zpráv spočívá v ověření totožnosti jednotlivých komunikujících entit. Směrovací protokol OSPFv3 umožňuje autentizovat své režijní zprávy za pomoci protokolu IPsec. Způsob autentizace může být buď globální, v rámci určité oblasti, nebo v rámci konkrétních rozhraní. IPsec je protokol síťové vrstvy, který slouží k autentizování a šifrování komunikace. Protokol IPsec je v podstatě seskupení více protokolů, které slouží k zabezpečení sítě. Pomocí IPsec můžeme vytvořit například zabezpečený tunel v rámci nezabezpečené sítě jakou je internet. Cisco IOS firewall je stavový firewall, který jsme schopni implementovat na samotném směrovači. Firewall povoluje provoz z vnější sítě na základě relace, která byla iniciována z vnitřní sítě. Nabízí možnosti ochrany sítě proti škodlivému kódu a jiným bezpečnostním rizikům.

A.3.4 Úkol 1 – Základní konfigurace sítě

a) Jako první nakonfigurujeme IPv6 adresaci na jednotlivých rozhraních R1, R2, R3 a DHCPv6 serveru dle obrázku A.31. IPv6 adresy se konfigurují obdobně jako IPv4 s tím rozdílem, že přidáváme klíčové slovo IPv6 a masku sítě píšeme v desítkovém tvaru.

```
R1(config-if)#ipv6 address adresa_sítě/prefix
```

Následuje ukázka konfigurace IPv6 adresy na rozhraní směrovače R1.

```
R1(config)#interface fastEthernet0/1
R1(config-if)#ipv6 address 3000:ABC::1/64
```

Příkazem **show ipv6 interface brief** si zobrazíme výslednou konfiguraci rozhraní směrovače R1.

```
R1#show ipv6 interface brief
FastEthernet0/0        [up/up]
    FE80::C801:1FFF:FEA0:8
    3000:AB::2
FastEthernet0/1        [up/up]
    FE80::C801:1FFF:FEA0:6
    3000:ABC::1
<výstup zkrácen>
```

Z výše uvedeného výpisu si všimněte adresy, které začínají jako FE80. Jak se nazývají tyto adresy a co je jejich role? Příkazem **ping** zkontrolujte konektivitu bod–bod.

b) Příkazem **ipv6 unicast–routing** spustíme ipv6 směrování na směrovačích R1, R2, R3 a DHCPv6. Následuje ukázka použití tohoto příkazu.

```
R1(config)#ipv6 unicast-routing
```

c) V tomto kroku nakonfigurujte v topologií směrovací protokol OSPFv3 s číslem instance 1. V celé síti se bude nacházet pouze jedna oblast a to *area 0*. Spuštění instance OSPFv3 je možné dvěma způsoby. Jeden je za pomoci příkazu **ipv6 router ospf** *číslo_instance*.

```
R1(config)#ipv6 router ospf 1
```

Druhý způsob je v rámci konfiguračního režimu rozhraní, na kterém je zároveň dána síť, která se příkazem zařadí do směrovacího procesu.

```
R1(config)#interface fastEthernet0/0
R1(config-if)#ipv6 ospf 1 area 0
```

Po příkazu **ipv6 router ospf 1** se spustí OSPFv3 proces s číslem 1 a zároveň se na konzoli směrovače zobrazí následující výpis.

```
R1(config)#ipv6 router ospf 1
R1(config-rtr)#
*Mar 14 16:03:04.451: %OSPFv3-4-NORTRID: Process OSPFv3-1-IPv6
could not pick a router-id, please configure manually
```

Na směrovačích nefiguruje žádná IPv4 adresa. *Router ID* je vždy 32 bitová hodnota, která se odvozuje z adresy rozhraní loopback nebo fyzické adresy. Další varianta je nastavit *Router ID* přímo v režimu směrovacího protokolu. Na směrovačích nastavte *Router ID* následovně.

```
R1(config)#ipv6 router ospf 1
R1(config-rtr)#router-id 1.1.1.1
R2(config)#ipv6 router ospf 1
R2(config-rtr)#router-id 2.2.2.2
R3(config)#ipv6 router ospf 1
R3(config-rtr)#router-id 3.3.3.3
DHCPv6(config)#ipv6 router ospf 1
DHCPv6(config-rtr)#router-id 4.4.4.4
```

d) Jeden ze způsobů jak ověřit správnost konfigurace je zobrazit směrovací tabulku a skontrolovat cesty k daným sítím. Na toto nám poslouží příkaz **show ipv6 route ospf**.

```
R1#show ipv6 route ospf
<výstup zkrácen>
O   3000:A::/64 [110/2]
       via FE80::C803:1DFF:FE20:8, FastEthernet0/0
O   4000:AB::/64 [110/2]
       via FE80::C803:1DFF:FE20:8, FastEthernet0/0
O   4000:ABC::/64 [110/2]
       via FE80::C803:1DFF:FE20:8, FastEthernet0/0
<výstup zkrácen>
```

Z výše uvedeného výpisu vidíme, že směrovač R1 má ve své směrovací tabulce cesty ke každé síti v dané topologií. Samostatně zkontrolujte směrovací tabulky i pro zbývající směrovače. Zkontrolujte konektivitu v síti příkazem **ping** včetně PC1–PC2, PC1–DHCPv6 a PC2–DHCPv6.

A.3.5 Úkol 2 – Konfigurace DHCP serveru

a) Přistoupíme ke konfiguraci bezstavového DHCPv6, který je standardně využíván v rámci IPv6 sítí. V konfiguračním režimu směrovače DHCPv6 nejdříve vytvoříme DHCP pool s názvem IPV6. Tímto se zároveň dostaneme do režimu konfigurace DHCPv6.

```
Server(config)#ipv6 dhcp pool IPV6
Server(config-dhcpv6)#
```

b) Dále si nastavíme adresy DNS serverů a doménového jména.

```
Server(config-dhcpv6)#dns-server AAAA:BBBB::1
Server(config-dhcpv6)#dns-server CCCC:DDDD::1
Server(config-dhcpv6)#domain-name IPv6_example.com
```

c) Dalším krokem je aktivování DHCPv6 serveru. Toto se provede příkazem v rámci konfiguračního režimu rozhraní.

```
Server(config)#interface fastEthernet0/0
Server(config-if)#ipv6 dhcp server IPV6
```

Kde IPV6 reprezentuje náš DHCPv6 pool.

d) Na směrovači R1 rozhraní fastEthernet0/1 zadáme příkaz pro zapnutí zasílání tzv. NDP parametrů a zároveň nakonfigurujeme relay agenta.

```
R1(config)#interface fastEthernet0/1
R1(config-if)#ipv6 nd other-config-flag
R1(config-if)#ipv6 dhcp relay destination 3000:A::1
```

Parametr *other–config–flag* slouží k získání parametrů jako DNS adresy přes bezestavové DHCPv6.

e) Provedeme restart rozhraní PC2, které je připojeno do naší simulované sítě. K tomuto využijeme příkazy pro příkazový řádek systému Windows.

```
netsh interface set interface name="Připojení k místní síti"
admin=disabled
```

```
netsh interface set interface name="Připojení k místní síti"
admin=enabled
```

V rámci virtualizované koncové stanici PC2 si příkazem **ipconfig /all** zobrazíme adresní parametry rozhraní. Obrázek A.32 poukazuje na fakt, že adresa není zapůjčena a DHCPv6 server byl použit pouze na poskytnutí parametrů jako doménové jméno a DNS adresy.

```
C:\Windows\System32>ipconfig /all
Adaptér sítě Ethernet Připojení k místní síti:

   Přípona DNS podle připojení . . . : IPv6_example.com
   Popis . . . . . . . . . . . . . . : Intel(R) PRO/1000 MT - síťové připojení
   Fyzická Adresa. . . . . . . . . . : 00-0C-29-CC-90-08
   Protokol DHCP povolen . . . . . . : Ano
   Automatická konfigurace povolena  : Ano
   IPv6 adresa. . . . . . . . . . . . : 3000:abc::e1b4:cc9b:e1b4:63bb(Preferované)

   Dočasná IPv6 adresa. . . . . . . . : 3000:abc::bd59:7c1d:d8ee:3123(Preferované)
   Místní IPv6 adresa v rámci propojení . . . : fe80::e1b4:cc9b:e1b4:63bb%14(Pre
ferované)
   Výchozí brána . . . . . . . . . . . : fe80::c803:fff:fe30:0%14
   IAID DHCPv6  . . . . . . . . . . . : 285215785
   DUID klienta DHCPv6. . . . . . . . : 00-01-00-01-1C-4B-19-57-00-0C-29-CC-90-08

   Servery DNS . . . . . . . . . . . : aaaa:bbbb::1
                                       cccc:dddd::1
   Rozhraní NetBios nad protokolem TCP/IP. . . . . . . . : Povoleno
   Seznam hledání přípon DNS specifických pro připojení:
                                       IPv6_example.com
```

Obr. A.32: Výpis adresních parametrů stanice PC2 (bezstavový DHCPv6).

f) Dále nakonfigurujeme na servery služby jako HTTP a HTTPS.

```
Server(config)#ip http server
Server(config)#ip http secure-server
```

Po zadání příkazu **ip http secure–server** se vygeneruje 1024 bitový RSA klíč jak je vidět na následujícím výpisu.

```
% Generating 1024 bit RSA keys, keys will be non-exportable...
[OK] (elapsed time was 1 seconds)

*Mar 21 16:00:04.051: %SSH-5-ENABLED: SSH 1.99 has been enabled
Server(config)#
*Mar 21 16:00:04.167: %PKI-4-NOAUTOSAVE: Configuration was modified.
Issue "write memory" to save new certificate
```

A.3.6 Úkol 3 – Zabezpečení OSPFv3

a) I v případě OSPFv3 můžeme řešit autentizaci směrovacích zpráv globálně, v rámci celé oblasti, nebo na konkrétním rozhraní. Nakonfigurujeme autentizaci mezi směrovačem R2 a R3 v rámci jejich rozhraní. Následující příkazy aplikujte na obě zařízení.

```
R2(config)#interface fastEthernet1/0
R2(config-if)#ipv6 ospf authentication ipsec spi 256
md5 12345678901234567890123456789012
```

Zobrazíme si stav autentizace na daném rozhraní. Následuje zkrácený výpis.

```
R3#show ipv6 ospf interface
<výstup zkrácen>
MD5 Authentication SPI 256, secure socket state UP (errors: 0)
<výstup zkrácen>
```

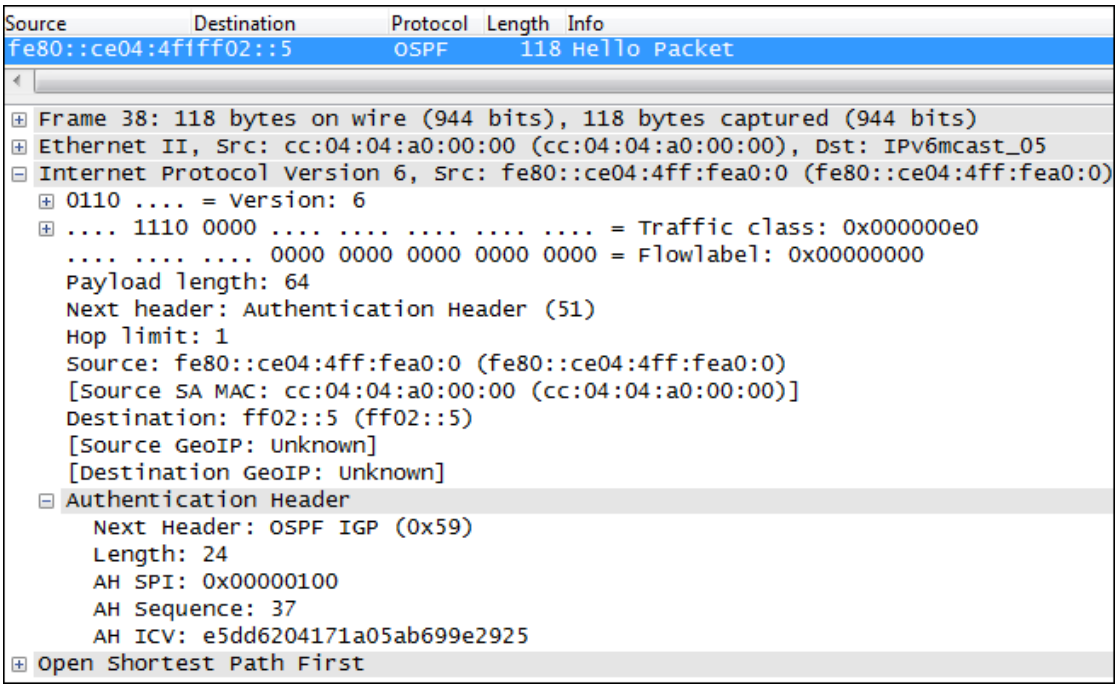
Stejný výpis bychom měli obdržet na směrovači R2.

b) Na směrovači R3 si zapněte zachytávání provozu na rozhraní fastEthernet0/0. Měli byste být schopni zachytit provoz, který ilustruje obrázek A.33.

Source	Destination	Protocol	Length	Info
fe80::ce04:4fiff02::5		OSPF	118	Hello Packet
fe80::c803:1diff02::5		OSPF	118	Hello Packet
fe80::ce04:4fiff02::5		OSPF	118	Hello Packet
fe80::c803:1diff02::5		OSPF	118	Hello Packet
fe80::ce04:4fiff02::5		OSPF	118	Hello Packet

Obr. A.33: Zachycený provoz autentizovných hello paketů OSPFv3.

Obrázek A.34 zachycuje autentizovaný hello paket. Z výpisu je vidět, že pole *Next Header* hlavičky IPv6, nese hodnotu 51, která indikuje autentizační hlavičku. Zachycený provoz si samostatně prostudujte a všimněte si, že v případě IPsec se řeší autentizace mimo hlavičky OSPFv3 paketu. V případě paketů OSPFv2 se využívalo speciální pole pro autentizaci.



Obr. A.34: Autentizované hello pakety protokolu OSPFv3.

c) Další užitečný příkaz v rámci IPsec a OSPFv3 je **show crypto ipsec sa**. Na následujícím výpisu si všimněte parametry SPI (Security Parameter Index) a Transform. Hodnota 256 představuje námi nastaven parametr SPI a v případě parametru Transform se využíva hešovací algoritmus MD5.

```
R3#show crypto ipsec sa
interface: FastEthernet0/0
<výstup zkrácen>
   IPsecv6 policy name: OSPFv3-1-256
   IPsecv6-created ACL name: FastEthernet0/0-ipsecv6-ACL
<výstup zkrácen>
     inbound esp sas:

     inbound ah sas:
      spi: 0x100(256)
        transform: ah-md5-hmac,
<výstup zkrácen>
```

d) Příkazem **show crypto ipsec policy** si zobrazíme námi nastavený IPsec klíč.

```
R2#show crypto ipsec policy
Crypto IPsec client security policy data

Policy name:      OSPFv3-256
Policy refcount:  1
Inbound  AH SPI:  256 (0x100)
Outbound AH SPI:  256 (0x100)
Inbound  AH Key:  12345678901234567890123456789012
Outbound AH Key:  12345678901234567890123456789012
Transform set:    ah-md5-hmac
```

A.3.7 Úkol 4 – Konfigurace IPv6 IPsec tunelu

a) V našem případě budeme považovat síť mezi R2 a R3 za nedůvěryhodnou. V tomto kroku nakonfigurujeme jaké parametry se použijí pro první fázi IKE politiky. Na R2 a R3 proveďte stejné kroky.

```
R3(config)#crypto isakmp policy 10
R3(config-isakmp)#encryption 3des
R3(config-isakmp)#group 2
R3(config-isakmp)#authentication pre-share
```

Skupinové číslo group označuje velikost modula pro Diffie–Hellman výměnu klíčů. Typ použitého šifrování jsme zvolili 3DES (Triple Data Encryption Standard). Na směrovačích nastavíme klíč *SEC* a přiřadíme k nim daného souseda.

```
R3(config)#crypto isakmp key 0 SEC address ipv6 4000:AB::1/64
R2(config)#crypto isakmp key 0 SEC address ipv6 4000:AB::2/64
```

b) Ve druhém kroku nakonfigurujeme IPsec transform set. Tím nakonfigurujeme jaké parametry se použijí pro druhou fázi IKE politiky. Dále nakonfigurujeme IPsec profil. Na směrovačích R2 a R3 proveďte stejné kroky.

```
R3(config)#crypto ipsec transform-set IPV6TRANSFORM esp-3des
esp-sha-hmac
R3(cfg-crypto-trans)#mode tunnel
R3(cfg-crypto-trans)#exit
R3(config)#crypto ipsec profile IPSECPROFILE
R3(ipsec-profile)#set transform-set IPV6TRANSFORM
```

c) Ve třetím kroku nakonfigurujeme ISAKMP profil s názvem *3DESISAKMP* v rámci IPv6, přičemž adresa identity musí ukazovat na správného souseda. Na R2 a R3 proveďte stejné kroky až na adresu souseda. V případě R2 to bude 4000:AB::2/64.

```
R3(config)#crypto isakmp profile 3DESISAKMP
R3(conf-isa-prof)#self-identity address ipv6
R3(conf-isa-prof)#match identity address ipv6 4000:AB::1/64
R3(conf-isa-prof)#keyring default
```

d) V dalším kroku vytvoříme statický tunel, který bude pracovat v režimu *ipsec ipv6*. To znamená, IPsec provoz v rámci protokolu IPv6. Adresa tunelů bude z rozsahu 5000::/64. Stejné kroky aplikujte na směrovačích R2 a R3, přičemž na R2 bude adresa tunelu 5000 :: 1/64 a patřičně upravte zdrojovou a cílovou adresu tunelu.

```
R3(config)#interface tunnel 0
R3(config-if)#ipv6 address 5000::2/64
R3(config-if)#tunnel source 4000:AB::2
R3(config-if)#tunnel destination 4000:AB::1
R3(config-if)#tunnel mode ipsec ipv6
R3(config-if)#tunnel protection ipsec profile IPSECPROFILE
```

Následně byste v konzole měli obdržet následující výpis.

```
*Mar  1 08:13:16.709: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is ON
```

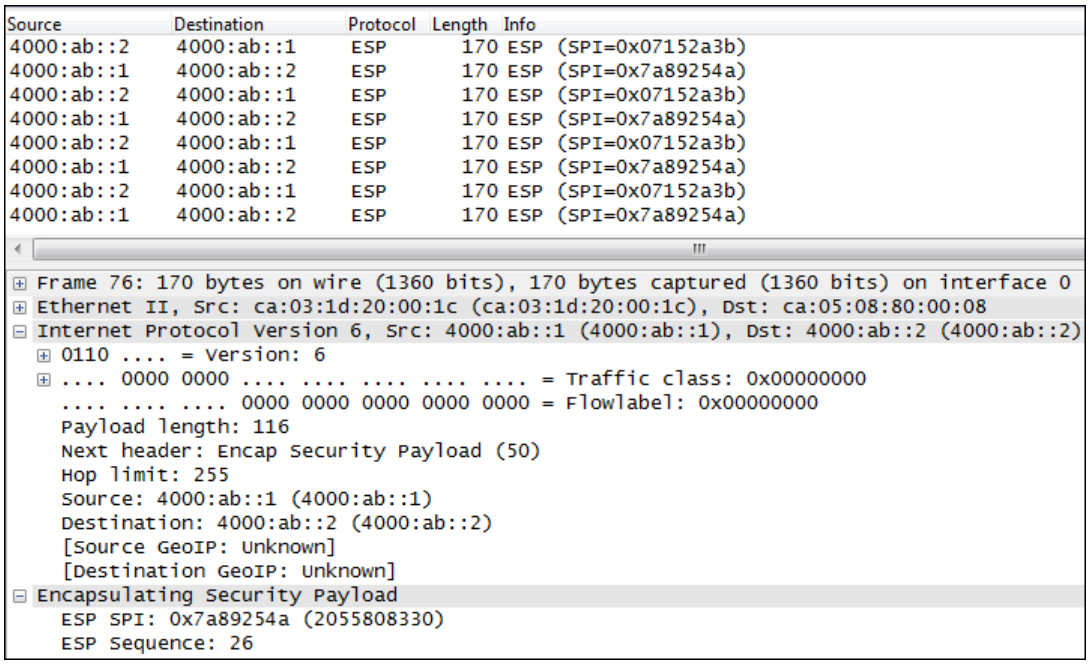
e) Na směrovači R3 vypněte směrovací protokol OSPFv3 a zároveň vytvořte výchozí cestu pro pakety směrující ze sítě 4000:ABC::/64 tak aby si PC1 zachovalo konektivitu se zbytkem sítě. Odchozí rozhraní nakonfigurujte jako rozhraní tunelu.

```
R3(config)#no ipv6 router ospf 1
R3(config)#ipv6 route ::/0 tunnel 0
```

Na směrovači R2 vytvořte statickou cestu do sítě 4000:ABC::/64, přičemž odchozí rozhraní bude rozhraní tunelu. Tuto cestu redistribujte do zbytku sítě.

```
R2(config)#ipv6 route 4000:ABC::/64 tunnel 0
R2(config)#ipv6 router ospf 1
R2(config-rtr)#redistribute static
```

f) Na směrovači R3 rozhraní fastEthernet0/0 spusťte zachytávání provozu. Z virtualizované stanice PC1 se pomocí příkazu **ping** dotazujte na adresu DHCPv6 serveru. Měli byste být schopni zachytit provoz jako zobrazuje obrázek A.35. Z výše uvedeného obrázku je vidět, že zdrojová a cílová adresa paketů je adresa samotného tunelu. V tomto případě se datová část paketu zašifruje a figuruje jako ESP. Dále můžeme vidět v hlavičce IPv6 paketu v poli *Next header* hodnotu 50, která indikuje právě ESP.



```
Source          Destination      Protocol  Length  Info
4000:ab::2      4000:ab::1       ESP        170 ESP (SPI=0x07152a3b)
4000:ab::1      4000:ab::2       ESP        170 ESP (SPI=0x7a89254a)
4000:ab::2      4000:ab::1       ESP        170 ESP (SPI=0x07152a3b)
4000:ab::1      4000:ab::2       ESP        170 ESP (SPI=0x7a89254a)
4000:ab::2      4000:ab::1       ESP        170 ESP (SPI=0x07152a3b)
4000:ab::1      4000:ab::2       ESP        170 ESP (SPI=0x7a89254a)
4000:ab::2      4000:ab::1       ESP        170 ESP (SPI=0x07152a3b)
4000:ab::1      4000:ab::2       ESP        170 ESP (SPI=0x7a89254a)

⊞ Frame 76: 170 bytes on wire (1360 bits), 170 bytes captured (1360 bits) on interface 0
⊞ Ethernet II, Src: ca:03:1d:20:00:1c (ca:03:1d:20:00:1c), Dst: ca:05:08:80:00:08
⊟ Internet Protocol Version 6, Src: 4000:ab::1 (4000:ab::1), Dst: 4000:ab::2 (4000:ab::2)
  ⊞ 0110 .... = Version: 6
  ⊞ .... 0000 0000 .... .... .... .... .... = Traffic class: 0x00000000
    .... .... .... 0000 0000 0000 0000 0000 = Flowlabel: 0x00000000
    Payload length: 116
    Next header: Encap Security Payload (50)
    Hop limit: 255
    Source: 4000:ab::1 (4000:ab::1)
    Destination: 4000:ab::2 (4000:ab::2)
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
⊟ Encapsulating Security Payload
    ESP SPI: 0x7a89254a (2055808330)
    ESP Sequence: 26
```

Obr. A.35: Zachycený provoz ICMPv6 v rámci IPsec tunelu.

A.3.8 Úkol 5 – Konfigurace IPv6 přístupových seznamů

Konfigurace IPv6 přístupových seznamů se řeší obdobně jako tomu bylo u verze IPv4. Změna nastala v syntaxi příkazů. První přístupový seznam si nakonfigurujeme na směrovači R3. Jeho úkolem bude zakázat ICMPv6 echo požadavek a odpovědi do sítě za směrovač R3 rozhraní fastEthernet0/0.

a) Jako první zkontrolujeme konektivitu mezi PC1 a zbytkem sítě. Příkazem **ping** otestujte konektivitu z PC1 na adresu serveru 3000:A::1. Ping by měl proběhnout úspěšně.

b) V tomto kroku nakonfigurujeme přístupový seznam který bude mít název *DENY–ICMPV6–PING*. Tento seznam bude obsahovat s třemi záznamy.

```
R3(config)#ipv6 access-list DENY-ICMPV6-PING
R3(config-ipv6-acl)#deny icmp any any echo-request
R3(config-ipv6-acl)#deny icmp any any echo-reply
R3(config-ipv6-acl)#permit ipv6 any any
```

Standardně se na konci každého přístupového seznamu nachází záznam deny any any. V případě IPv6 se implicitně na konci každého přístupového seznamu nacházejí následující tři zaznamy.

```
permit icmp any any nd-na
permit icmp any any nd-ns
```

```
deny ipv6 any any
```

c) Dalším krokem bude aplikovat tento přístupový seznam na rozhraní fastEthernet0/1 směrovače R3. Na toto se použije příkaz **traffic–filter**.

```
R3(config)#interface fastEthernet0/1
R3(config-if)#ipv6 traffic-filter DENY-ICMPV6-PING in
```

d) Nyní otestujte konektivitu příkazem **ping** ze stanice PC1 na server. Ping by měl být neúspěšný. Zároveň s příkazového řádku stanice PC1 otestujte konektivitu na port 80 příkazem **telnet** *3000:A::1 80*. Po úspěšném připojení na port 80 si zobrazíme seznam připojení na servery následujícím příkazem.

```
Server#show ip http server connection
HTTP server current connections:
local-ipaddress:port  remote-ipaddress:port in-bytes   out-bytes
[3000:A::1]:80
                      [4000:ABC::30A9:C381:49F5:1843]:49171
                                              0          0
```

Pokud budeme telnetem testovat port 443 měli bychom obdržet obdobný výpis jako je znázorněn níže.

```
Server#show ip http server connection
HTTP server current connections:
local-ipaddress:port  remote-ipaddress:port in-bytes   out-bytes
[3000:A::1]:443
                      [4000:ABC::30A9:C381:49F5:1843]:49172
                                              0          0
```

Z výše uvedených výpisů a testování si můžeme být jisti, že jsme zablokovali jen ICMPv6 zprávy typu echo–request a echo–reply. V případě ICMPv6 protokolu musíme být obzvlášť opatrní, protože IPv6 je velmi úzce spojen právě s protokolem ICMPv6. U většiny přístupových seznamů musíme povolit i ICMPv6 zprávy typu 2 *Packet Too Big*. Tyto zprávy jsou charakteristické pro oznámení MTU.

A.3.9 Úkol 6 – Konfigurace IPv6 Cisco IOS firewallu

V této části úkolu nakonfigurujeme stavový firewall na směrovači R2. Jedním z úkolů stavového firewallu může být zahazování provozu, který pochází z vnější sítě, přičemž výjimku tvoří provoz, který byl iniciován z vnitřní sítě. Náš firewall bude sledovat relace, které byly iniciovány ze sítě 3000:ABC::/64 do sítě 3000:A::/64.

a) Jako první vytvoříme stavový firewall s názvem IPV6–FIREWALL, který bude sledovat relace protokolů TCP, UDP a ICMP.

```
R2(config)#ipv6 inspect name IPV6-FIREWALL tcp
R2(config)#ipv6 inspect name IPV6-FIREWALL udp
R2(config)#ipv6 inspect name IPV6-FIREWALL icmp
```

b) V tomto kroku vytvoříme přístupový seznam s názvem *WWW*, který bude povolovat ICMPv6 provoz, protokol HTTP, HTTPS a DHCPv6. Dále bude seznam obsahovat záznam, který bude povolovat provoz lokální linkové adresy.

```
R2(config)#ipv6 access-list WWW
R2(config-ipv6-acl)#permit icmp any any
R2(config-ipv6-acl)#permit tcp 3000:ABC::/64 3000:A::/64 eq www
R2(config-ipv6-acl)#permit tcp 3000:ABC::/64 3000:A::/64 eq 443
R2(config-ipv6-acl)#permit udp 3000:ABC::/64 3000:A::/64 eq 547
R2(config-ipv6-acl)#permit ipv6 FE80::/10 any
R2(config-ipv6-acl)#deny ipv6 any any
```

c) Dále vytvoříme přístupový seznam s názvem *DENY–ALL*, který bude povolovat jen minimální nutný provoz, který je důležitý aby se zachovala konektivita.

```
R2(config)#ipv6 access-list DENY-ALL
R2(config-ipv6-acl)#permit icmp any any nd-na
R2(config-ipv6-acl)#permit icmp any any nd-ns
R2(config-ipv6-acl)#permit icmp any any packet-too-big
R2(config-ipv6-acl)#deny icmp any any
R2(config-ipv6-acl)#permit ipv6 FE80::/16 any
R2(config-ipv6-acl)#deny ipv6 any any
```

d) Jako poslední přiřadíme přístupové seznamy a stavový firewall na správné rozhraní směrovače R2.

```
R2(config)#interface fastEthernet0/0
R2(config-if)#ipv6 inspect IPV6-FIREWALL in
R2(config-if)#ipv6 traffic-filter WWW in

R2(config)#interface fastEthernet0/1
R2(config-if)#ipv6 traffic-filter DENY-ALL in
```

e) V tomto bodě otestujeme nakonfigurovaný firewall. Z koncové stanice PC2 si otevřete tři příkazové řádky. Z jednoho se dotazujte příkazem **ping** na adresu serveru 3000:A::1 a z dalších otestujte otevřené porty 80 a 443 pomocí příkazu **telnet** nasledovně:

ping *3000:A::1* **-t**

telnet *3000:A::1 80*

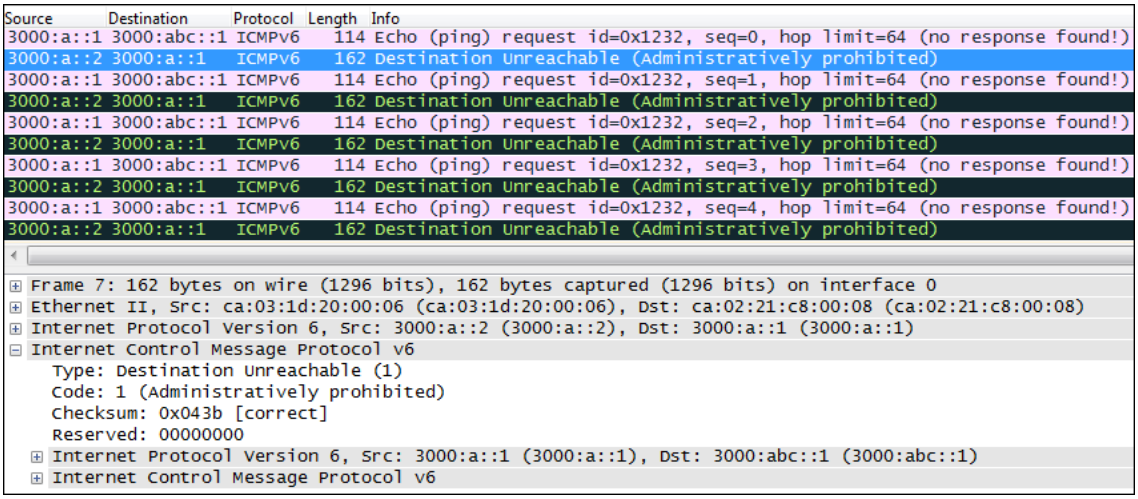
telnet *3000:A::1 443*

Zároveň si na směrovači R2 zobrazíme seznam otevřených relací. Použijeme na to příkaz **show ipv6 inspect sessions**.

```
R2#show ipv6 inspect sessions
Established Sessions
 Session 6B0C3014 (3000:ABC::741B:CDBF:790F:44DC:49160)=>
 (3000:A::1:80) tcp SIS_OPEN
 Session 6B0C31D4 (3000:ABC::741B:CDBF:790F:44DC:49159)=>
 (3000:A::1:443) tcp SIS_OPEN
 Session 6B0C2E54 (3000:ABC::741B:CDBF:790F:44DC:0)=>
 (3000:A::1:0) icmp SIS_OPEN
```

f) Nyní si spustíme zachytávání provozu na rozhraní serveru fastEthernet0/0. Ze serveru se pomocí příkazu **ping** dotazujte na adresu 3000:ABC::1. Všimněte si indikaci neúspěšnosti příkazu v konzoli směrovače. Měli byste obdržet výpis jako je uveden níže. Zároveň byste měli zachytit provoz jako je ilustrován na obrázku A.36. Všimněte si hlavně typ a kód ICMPv6 zprávy.

```
Server#ping 3000:ABC::1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 3000:ABC::1, timeout is 2 seconds:
AAAAA
Success rate is 0 percent (0/5)
```



Obr. A.36: Zachycený provoz dotazu ping na servery.

g) Na směrovači R2 spustíme příkazy **debug ipv6 inspect object–creation** a **debug ipv6 inspect icmp** výpis zpracování ICMP provozu stavovým firewallem. Z koncové stanice PC2 se příkazem **ping** dotazujte na adresu serveru 3000:A::1. Následující řádky poukazují zachycený výpis.

```
FIREWALL ipv6_insp_init_sis - OBJ_CREATE: create sis 6B10A074
FIREWALL ipv6_insp_init_sis: - Pak 68E8EFF4 sis 6B10A074 initiator_addr
(3000:ABC::407A:42CD:A700:AF19:0) responder_addr (3000:A::1:0)
FIREWALL icmp_info created: 0x6B0561B8
FIREWALL OBJ-CREATE: sid 6B116918 acl DENY-ALL Prot: icmp
Src 3000:A::1 Port [0]
Dst 3000:ABC::407A:42CD:A700:AF19 Port [129]
FIREWALL OBJ-CREATE: sid 6B1168B0 acl DENY-ALL Prot: icmp
Src 3000:A::1 Port [0]
Dst 3000:ABC::407A:42CD:A700:AF19 Port [2]
FIREWALL OBJ-CREATE: sid 6B116848 acl DENY-ALL Prot: icmp
Src 3000:A::1 Port [0]
Dst 3000:ABC::407A:42CD:A700:AF19 Port [1]
FIREWALL OBJ-CREATE: sid 6B1167E0 acl DENY-ALL Prot: icmp
Src 3000:A::1 Port [0]
Dst 3000:ABC::407A:42CD:A700:AF19 Port [3]
FIREWALL* sis 6B10A074 L4 inspect result: PASS packet 68E8EFF4
(3000:ABC::407A:42CD:A700:AF19:0) (3000:A::1:0) bytes 40 icmp
FIREWALL* sis 6B10A074 L4 inspect result: PASS packet 68E8F4C0
(3000:A::1:0) (3000:ABC::407A:42CD:A700:AF19:0) bytes 40 icmp
<výstup zkrácen>
```

Předchozí řádky poukazují na vytvoření objektu podle, kterého směrovač propouští provoz, který byl iniciován z vnitřní sítě.

A.3.10 Kontrolní otázky

1. Jakými dvěma způsoby můžeme aktivovat autentizaci zpráv OSPFv3?
2. Jak se liší způsob autentizace OSPFv3 zpráv oproti OSPFv2?
3. Jaký je princip IPsec tunelu a kde se dá využít toto zabezpečení komunikace?
4. Jaký je význam přístupových seznamů?
5. Jak se liší konfigurace přístupového seznamu na konkrétním rozhraní v rámci IPv4 a IPv6 (Berme v úvahu konfigurační příkazy Cisco)?
6. Jaké je implicitní ukončení přístupového seznamu v IPv6?
7. Jaký je princip stavového Cisco IOS firewallu?

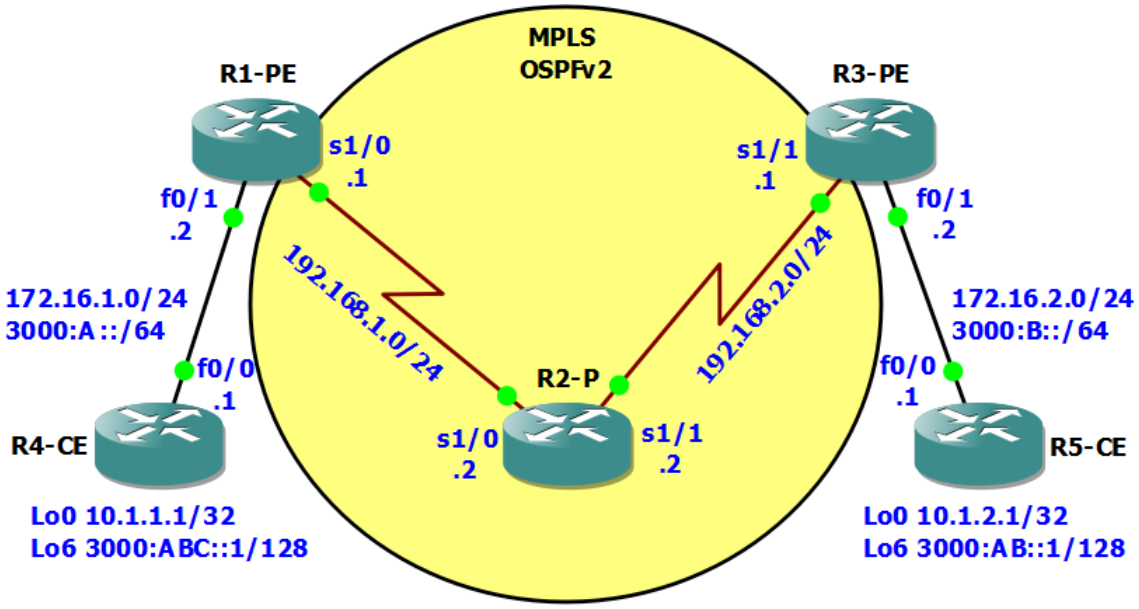
A.3.11 Samostatná úloha

Nakonfigurujte autentizaci OSPFv3 režijních zpráv v celé topologi. Pokuste se autentizaci řešit globálně pro danou oblast OSPFv3 a ne v rámci konkrétního rozhraní.

Poznámka: Přiložené DVD obsahuje konfiguraci zařízení před a po samostatné úloze. Dále obsahuje samotné zdrojové soubory projektů pro GNS3. Přiložené DVD neobsahuje virtualizované stanice v programu VMware. V případě další práce s laboratorními úlohami je možné využít i VirtualBox. V rámci virtualizovaných stanic Windows 7 není nutná žádná specifická konfigurace.

A.4 Laboratorní úloha 4

A.4.1 Zapojení topologie sítě



Obr. A.37: Zapojení topologie sítě.

A.4.2 Hlavní cíle úlohy

- Základní konfigurace MPLS
- Konfigurace MPLS autentizace
- Konfigurace MPLS a IPv6 pomocí tunelu 6to4
- Konfigurace MPLS a IPv6 pomocí techniky 6PE

A.4.3 Teoretický úvod

Použití MPLS v IP sítích je zejména kvůli rychlejšímu přepínání paketů na základě štítků. Tyto štítky obsahují číselní hodnotu, která je eqvivalentem cílové IP adresy. Informace podle kterých zařízení přiděluje hodnoty štítku jsou kombinací údajů ze síťové a vyšších vrstev. LDP protokol slouží k distribuci štítků a v rámci lepší kontroly a bezpečnosti podporuje autentizaci na transportní vrstvě pomocí hešovacího algoritmu MD5. MPLS v současnosti nativně nepodporuje IPv6 protokol. MPLS v rámci řídící roviny pracuje s protokoly IPv4. Řešení tohoto problému nabízí několik metod. Jedna z nich je použít tranzitní techniku tunelování typu 6to4 v rámci CE směrovačů. Druhá varianta je použít techniku, která se nazývá 6PE. Toto řešení se dotkne pouze hraničních zařízení MPLS sítě resp. jen PE směrovačů.

A.4.4 Úkol 1 – Základní konfigurace MPLS

Poznámka: Úloha má implicitně nakonfigurovanou IPv4 adresaci a v celé topologií je spuštění OSPFv2 směrovací protokol. Konfigurací IPv6 náležitostí se zabývají další části laboratorní úlohy.

a) Spuštění a základní nastavení MPLS protokolu je vcelku triviální záležitost. V oblasti, která je na obrázku A.37 barevně označena jako MPLS, nakonfigurujeme rozhraní směrovačů. Příkazem **mpls ip** spustíme dynamické zasílání MPLS zpráv. Příkaz **mpls label protocol ldp** slouží k nastavení typu protokolu pro výměnu MPLS zpráv. Na výběr je ještě protokol TDP nebo můžeme zvolit variantu kdy aktivujeme oba protokoli současně.

```
R1-PE(config)#interface serial1/0
R1-PE(config-if)#mpls ip
R1-PE(config-if)#mpls label protocol ldp

R3-PE(config)#interface serial1/1
R3-PE(config-if)#mpls ip
R3-PE(config-if)#mpls label protocol ldp
```

Na směrovači R2–P využijeme směrovací protokol OSPFv2. V konfiguračním režimu směrovacího procesu aktivujeme protokol LDP globálně na všechny rozhraní, které spadají do OSPFv2 oblasti *area 0*.

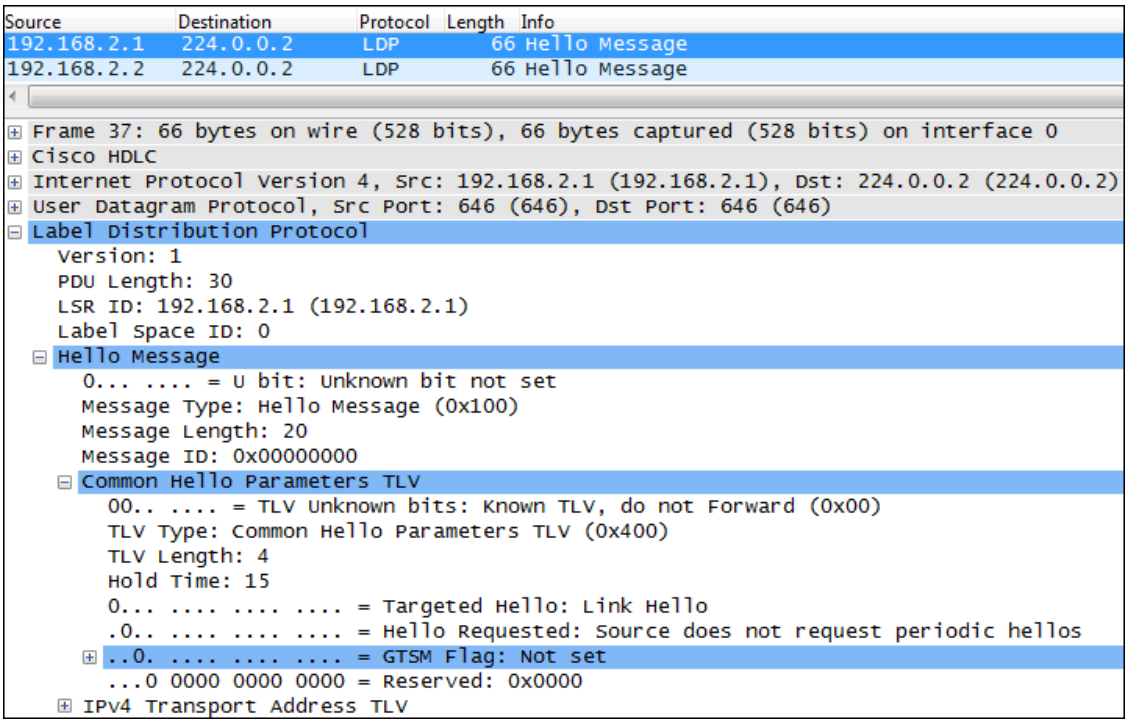
```
R2-P(config)#router ospf 1
R2-P(config-router)#mpls ldp autoconfig area 0
```

Po aplikování příkazu **mpls ldp autoconfig area** *0* již není nutné spouštět dynamické zasílání MPLS zpráv příkazem **mpls ip**. S příslušným příkazem si zkontrolujeme nastavení LDP protokolu v rámci směrovacího procesu OSPFv2.

```
R2-P#show ip ospf mpls ldp interface serial1/1
Serial1/1
  Process ID 1, Area 0
  LDP is configured through LDP autoconfig
  LDP-IGP Synchronization : Not required
  Holddown timer is disabled
  Interface is up
```

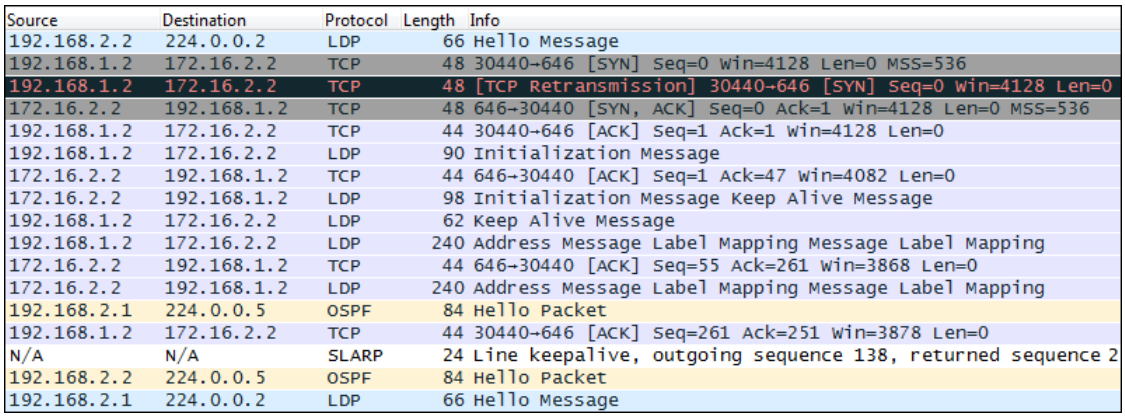
b) Spustíme si zachytávání provozu na směrovači R2–P rozhraní Serial1/1. Měli bychom být schopni zachytit LDP správy jako ilustruje obrázek A.38. Z obrázku je vidět, že se jedná o verzi protokolu 1. ID LSR směrovače je 192.168.2.1

takže se jedná o paket, který byl odeslán ze směrovače R3–PE. K zasílání hello zpráv se využívá UDP protokol se zdrojovým a cílovým portem 646, přičemž je zpráva zasílána na skupinovou adresu 224.0.0.2.



Obr. A.38: Zachycené hello zprávy protokolu LDP.

Příkazem **clear mpls ldp neighbor *** na směrovači R3–PE vyvoláme restartování sousedství MPLS. Obrázek A.39 ilustruje zachycený provoz LDP protokolu v rámci navázání spojení na směrovači R2–P rozhraní Serial1/1. Z obrázku je vidět, že k navázání relace se využívá TCP protokol s číslem portu 646.



Obr. A.39: Navázání sousedství v rámci protokolu LDP.

c) Na příslušných směrovačích si zobrazíme bližší informace o MPLS prostřednictvím příkazu **show mpls ldp neighbor**. Konkrétně na směrovači R2–P bychom měli obdržet následující výpis.

```
R2-P#show mpls ldp neighbor
    Peer LDP Ident: 172.16.1.2:0; Local LDP Ident 192.168.1.2:0
        TCP connection: 172.16.1.2.646 - 192.168.1.2.11543
        State: Oper; Msgs sent/rcvd: 103/98; Downstream
        Up time: 01:17:02
        LDP discovery sources:
          Serial1/0, Src IP addr: 192.168.1.1
        Addresses bound to peer LDP Ident:
          172.16.1.2      192.168.1.1
    Peer LDP Ident: 172.16.2.2:0; Local LDP Ident 192.168.1.2:0
        TCP connection: 172.16.2.2.646 - 192.168.1.2.30440
        State: Oper; Msgs sent/rcvd: 71/70; Downstream
        Up time: 00:53:59
        LDP discovery sources:
          Serial1/1, Src IP addr: 192.168.2.1
        Addresses bound to peer LDP Ident:
          172.16.2.2      192.168.2.1
```

Z výpisu jsme obdrželi informace jako identifikace souseda, TCP spojení mezi lokálním a sousedním směrovačem a k nim příslušející čísla portů, délku trvání spojení, rozhraní přes které jsou LDP zprávy objevovány. Dále si na směrovačích R1–PE a R2–P příkazem **show mpls forwarding–table** zobrazíme LFIB tabulku. Následuje ukázka výpisu.

```
R1-PE#show mpls forwarding-table
Local   Outgoing  Prefix          Bytes Label  Outgoing  Next Hop
Label   Label     or Tunnel Id    Switched     interface
16      Pop Label 192.168.2.0/24  0            Se1/0     point2point
17      16        172.16.2.0/24   0            Se1/0     point2point
18      18        10.1.2.1/32     0            Se1/0     point2point
19      No Label  10.1.1.1/32     0            Fa0/1     172.16.1.1
```

```
R2-P#show mpls forwarding-table
Local   Outgoing  Prefix          Bytes Label  Outgoing  Next Hop
Label   Label     or Tunnel Id    Switched     interface
16      Pop Label 172.16.2.0/24   0            Se1/1     point2point
17      Pop Label 172.16.1.0/24   0            Se1/0     point2point
18      18        10.1.2.1/32     0            Se1/1     point2point
19      19        10.1.1.1/32     0            Se1/0     point2point
```

Z výpisů je vidět, že číslování štítků začíná od hodnoty 16. Hodnoty 0–15 jsou rezervovány. *Pop Label* značí odstranění štítku a *No Label* značí, že nepřichází k přidělení štítku. Tabulka zobrazuje jako výstupní rozhraní tak i next hop parametr.

d) Na směrovači R2–P vypněte funkci CEF příkazem **no ip cef**.

```
R2-P#configure terminal
R2-P(config)#no ip cef
```

Následně si znovu zobrazte LFIB tabulku na směrovači R2–P. Měli byste vidět výpis jako je zobrazen níže.

```
R2-P#show mpls forwarding-table
Local   Outgoing  Prefix          Bytes Label  Outgoing   Next Hop
Label   Label     or Tunnel Id    Switched     interface
16      No Label  172.16.2.0/24   0            drop
17      No Label  172.16.1.0/24   0            drop
18      No Label  10.1.2.1/32     0            drop
19      No Label  10.1.1.1/32     0            drop
```

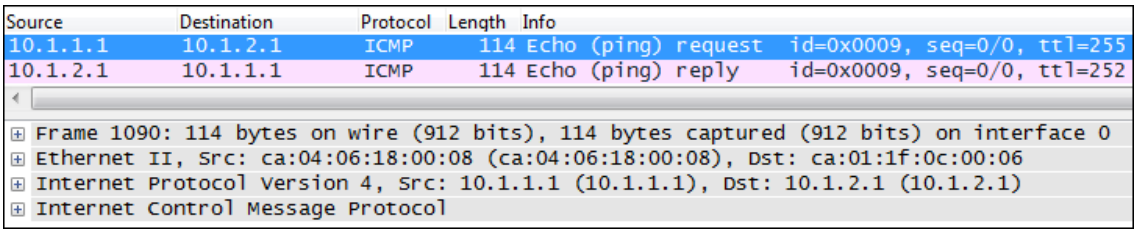
Z výpisu je patrné, že MPLS je nefunkční pokud není spuštěna funkce CEF. Technicky by to nebylo možné, protože LFIB tabulka je závislá na tabulce FIB. Znovu spusťte funkci CEF. Příkazem **show mpls ip binding** si na směrovači R2–P zobrazíme další informace o přiřazených štítcích k jednotlivým sítím.

```
R2-P#show mpls ip binding
  10.1.1.1/32
        in label:     19
        out label:    19        lsr: 192.168.1.1:0
        out label:    19        lsr: 192.168.2.1:0
  10.1.2.1/32
        in label:     18
        out label:    18        lsr: 192.168.2.1:0
        out label:    18        lsr: 192.168.1.1:0
<výstup zkrácen>
```

e) Spusťte si zachytávání provozu na směrovači R1–PE fastEthernet0/1, Serial1/0 a na směrovači R2–P Serial1/1. Následně na směrovači R4–CE se budeme dotazovat příkazem **ping** na adresu 10.1.2.1, přičemž zdroj bude loopback0.

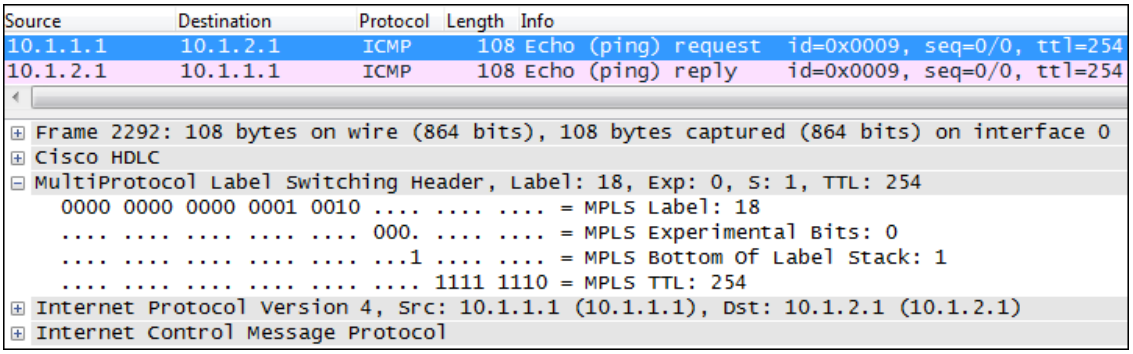
```
R4-CE#ping 10.1.2.1 source loopback0
```

Obrázek A.40 ilustruje dotaz ping zachycen na rozhraní směrovače R1–PE fastEthernet0/1. Jelikož na tomto rozhraní není nakonfigurováno MPLS nedochází k přidělení štítku.



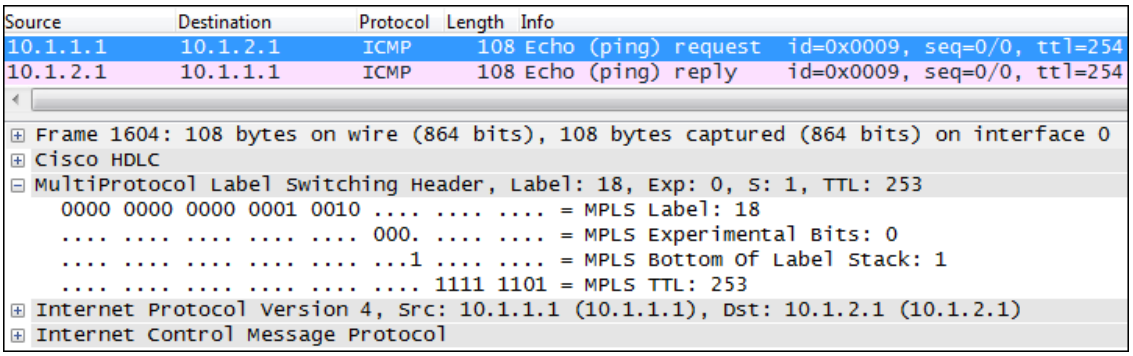
Obr. A.40: Zachycený dotaz příkazu ping, R1–PE rozhraní fastEthernet0/1.

Obrázek A.41 ilustruje dotaz ping zachycen na rozhraní směrovače R1–PE Serial1/0. V rámci tohoto rozhraní směrovač figuruje jako ELSR a primárně přidává a odstraňuje štítky. Konkrétně byla přiřazena hodnota 18 a hodnota MPLS TTL 254.



Obr. A.41: Zachycený dotaz příkazu ping, R1–PE rozhraní Serial1/0.

Obrázek A.42 ilustruje dotaz ping zachycen na rozhraní směrovače R2–P Serial1/1. Tento směrovač figuruje jako LSR a primárně přepíná štítky. Hodnota štítku byla ponechána a MPLS TTL se dekrementovalo na 253.



Obr. A.42: Zachycený dotaz příkazu ping, R2–P rozhraní Serial1/1.

Samostatně si zobrazte LFIB tabulky příslušných směrovačů a zkontrolujte hodnoty štítků v závislosti na zachyceném provozu.

A.4.5 Úkol 2 – Změna Router–ID MPLS

a) V tomto kroku nakonfigurujeme na směrovači R2–P nové rozhraní loopback následovně.

```
R2-P(config)#interface loopback50
R2-P(config-if)#ip address 50.50.50.50 255.255.255.255
```

b) Příkazem **show mpls ldp discovery** si zobrazíme aktuální nakonfigurované *Router ID* v rámci MPLS.

```
R2-P#show mpls ldp discovery
 Local LDP Identifier:
    192.168.2.2:0
<výstup zkrácen>
```

Jak je vidět z výše uvedeného výpisu, na směrovači R2–P konfigurací nového loopback rozhraní nenastala změna parametru *Router ID*.

c) Příkazem **mpls ldp router–id** nastavíme konkrétní loopback tak aby figuroval jako *Router ID*.

```
R2-P(config)#mpls ldp router-id loopback50
```

d) Po znovu zadání příkazu **show mpls ldp discovery** můžeme vidět, že *Router ID* se přesto nezměnilo.

```
R2-P#show mpls ldp discovery
 Local LDP Identifier:
    192.168.2.2:0
<výstup zkrácen>
```

Aby se změna projevila okamžitě po zadání příkazu je nutné použít klíčové slovo **force**.

```
R2-P(config)#mpls ldp router-id loopback50 force
```

Po tomto příkazu můžeme vidět následující výpis na konzoli.

```
%TDP-5-INFO: default: TDP ID removed
%LDP-5-NBRCHG: LDP Neighbor 192.168.1.1:0 (1) is DOWN (LDP Router
  ID changed)
%LDP-5-NBRCHG: LDP Neighbor 192.168.2.1:0 (2) is DOWN (LDP Router
  ID changed)
```

Po opětovném zadání příkazu **show mpls ldp discovery** můžeme vidět změnu parametru *Router ID*.

```
R2-P#show mpls ldp discovery
 Local LDP Identifier:
    50.50.50.50:0
<výstup zkrácen>
```

Zároveň si na směrovači zobrazte příkazem **show mpls ldp neighbor** aktuální stav sousedství v rámci MPLS. Můžeme se přesvědčit, že tabulka sousedství bude prázdná a stejně se ztratí sousedství mezi R1–PE a R3–PE. Je třeba si uvědomit, že nově přidaná adresa loopback nefiguruje v směrovacím procesu OSPFv2. Informace o této adrese se nešíří na zbývající směrovače.

e) Následujícími příkazy přidejte loopback do směrovacího procesu OSPFv2.

```
R2-P(config)#interface loopback 50
R2-P(config-if)#ip ospf 1 area 0
```

Zároveň na výstupu konzoly můžeme vidět následující zprávy.

```
%LDP-5-NBRCHG: LDP Neighbor 192.168.2.1:0 (1) is UP
%LDP-5-NBRCHG: LDP Neighbor 192.168.1.1:0 (2) is UP
```

Příkazem **show mpls ldp neighbor** si zobrazíme aktuální stav sousedství.

```
R2-P#show mpls ldp neighbor
    Peer LDP Ident: 192.168.2.1:0; Local LDP Ident 50.50.50.50:0
        TCP connection: 192.168.2.1.51701 - 50.50.50.50.646
    <výstup zkrácen>
    Peer LDP Ident: 192.168.1.1:0; Local LDP Ident 50.50.50.50:0
        TCP connection: 192.168.1.1.17681 - 50.50.50.50.646
    <výstup zkrácen>
```

f) Jako poslední krok nakonfigurujeme *Router ID* pro směrovače R1–PE a R3–PE. K tomuto využijeme fyzické rozhraní sériových linek směrovačů.

```
R1-PE(config)#mpls ldp router-id serial1/0 force
R3-PE(config)#mpls ldp router-id serial1/1 force
```

A.4.6 Úkol 3 – Konfigurace MPLS autentizace

Pro lepší kontrolu nad výměnou LDP zpráv v rámci MPLS protokolu můžeme využít autentizaci. Tato autentizace se řeší pomocí hešovacího algoritmu MD5 na transportní úrovni. V následující části úlohy nakonfigurujeme autentizaci LDP spojení mezí směrovači R2–P a R3–PE. Při použití autentizace je vhodné mít nastavené fixní *Router ID* buď prostřednictvím loopback nebo fyzického rozhraní. Toto jsme vyřešili v předchozích bodech.

a) Následujícím příkazem nakonfigurujeme autentizaci na směrovači R2–P.

```
R2-P(config)#mpls ldp neighbor 192.168.2.1 password MPLS_PASS
```

Adresa *192.168.2.1* reprezentuje *Router ID* parametr sousedního směrovače, v rámci kterého chceme nakonfigurovat autentizaci. *MPLS_PASS* představuje konkrétní heslo.

b) Obdobně nakonfigurujeme i směrovač R3–PE.

```
R3-PE(config)#mpls ldp neighbor 50.50.50.50 password MPLS_PASS
```

c) Příkazem **show mpls ldp neighbor password** si ověříme aktuální stav v rámci autentizace a stavu použití hesel na směrovači R2–P.

```
R2-P#show mpls ldp neighbor password
    Peer LDP Ident: 192.168.1.1:0; Local LDP Ident 50.50.50.50:0
        TCP connection: 192.168.1.1.56863 - 50.50.50.50.646
        Password: not required, none, in use
        State: Oper; Msgs sent/rcvd: 37/37
    Peer LDP Ident: 192.168.2.1:0; Local LDP Ident 50.50.50.50:0
        TCP connection: 192.168.2.1.64480 - 50.50.50.50.646
        Password: not required, neighbor, stale
        State: Oper; Msgs sent/rcvd: 36/37
```

Z výpisu můžeme vidět, že v rámci sousedství R2–P a R1–PE se používá režim *none* což značí, že není nastaveno žádné heslo. V rámci sousedství R2–P a R3–PE figuruje klíčové slovo *neighbor* a je ve stavu *stale*. Stav *stale* značí, že heslo se buď nepoužívá nebo je zastaralé v rámci sousedství.

d) K autentizaci dojde při restartování TCP spojení v rámci LDP sousedství směrovačů příkazem **clear mpls ldp neighbor ***. Další možností jak aktivovat autentizaci je vynutit použití hesla příkazem **mpls ldp password required**. Při této variantě si však musíme uvědomit, že směrovač bude požadovat heslo i od LDP sousedů mezi nimiž není nastaveno heslo. To znamená zrušení sousedství mezi těmi směrovači.

```
R2-P(config)#mpls ldp password required
```

Na směrovači R2–P byste měli zaznamenat výpis jako je zobrazen níže.

```
%LDP-5-NBRCHG: LDP Neighbor 192.168.2.1:0 (1) is DOWN
     (Session's MD5 password changed)
%LDP-5-NBRCHG: LDP Neighbor 192.168.1.1:0 (2) is DOWN
     (Session's MD5 password changed)
```

Příkaz na směrovači R2–P deaktivujte formou **no** a pro restartování spojení LDP použijte příkaz **clear mpls ldp neighbor ***.

```
R2-P(config)#no mpls ldp password required
R2-P#clear mpls ldp neighbor *
```

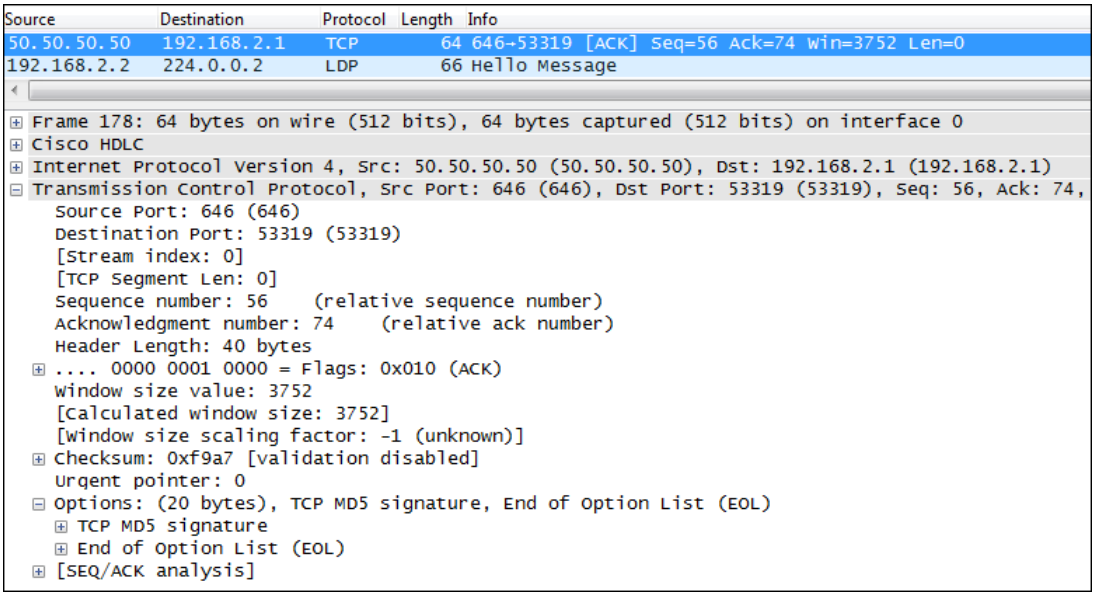
Na směrovači R2–P byste měli zaznamenat výpis jako je zobrazen níže.

```
%LDP-5-CLEAR_NBRS: Clear LDP neighbors (*) by console
%LDP-5-NBRCHG: LDP Neighbor 192.168.1.1:0 (3) is DOWN
     (User cleared session manually)
%LDP-5-NBRCHG: LDP Neighbor 192.168.2.1:0 (4) is DOWN
     (User cleared session manually)
```

Po opětovném zadání příkazu **show mpls ldp neighbor password** byste měli být schopni vidět parametry jako jsou zobrazeny ve výpisu níže. Mezi směrovači R2–P a R3–PE se vytvořilo LDP sousedství na základě hesla, které se i aktuálně používá.

```
R2-P#show mpls ldp neighbor password
   <výstup zkrácen>
    Peer LDP Ident: 192.168.2.1:0; Local LDP Ident 50.50.50.50:0
        TCP connection: 192.168.2.1.53319 - 50.50.50.50.646
        Password: not required, neighbor, in use
        State: Oper; Msgs sent/rcvd: 8/8
```

e) Na směrovači R2–P si spusťte zachytávání provozu na rozhraní Serial1/1. Na směrovači zadejte příkaz **clear mpls ldp neighbor ***. Měli byste být schopni zachytit provoz jako je zobrazen na obrázku A.43. Z ilustrace je vidět, že autentizace LDP protokolu se řeší na transportní vrstvě pomocí MD5 algoritmu.



Obr. A.43: Navázání LDP sousedství pomocí TCP spojení s autentizací.

A.4.7 Úkol 4 – MPLS a IPv6 pomocí tunelu 6to4

MPLS nativně nepodporuje IPv6. Jádro sítě MPLS pracuje s protokolovou sadou a směrovacími protokoly IPv4. Jedním ze způsobů jak vyřešit tento problém je zprovoznit tunel 6to4 v rámci komunikujících protilehlých IPv6 oblastí.

a) Nakonfigurujeme IPv6 adresaci podle obrázku A.37.

```
R4-CE(config)#interface loopback6
R4-CE(config-if)#ipv6 address 3000:ABC::1/128

R5-CE(config)#interface loopback6
R5-CE(config-if)#ipv6 address 3000:AB::1/128
```

Samostatně nakonfigurujte i linky mezi R4–CE, R1–PE a R5–CE, R3–PE.

b) Příkazem **ipv6 unicast–routing** spustíme ipv6 směrování na směrovačích R4–CE, R1–PE, R3–PE a R5–CE. Následuje ukázka použití tohoto příkazu.

```
R4-CE(config)#ipv6 unicast-routing
```

c) V rámci směrovačů R4–CE, R1–PE, R3–PE a R5–CE nakonfigurujeme směrovací protokol OSPFv3 s číslem instance 1. V celé síti se bude nacházet pouze jedna oblast a to *area 0*. Spuštění instance OSPFv3 je možné dvěma způsoby. Jeden je za pomoci příkazu **ipv6 router ospf** *číslo_instance*.

```
R4-CE(config)#ipv6 router ospf 1
```

Druhý způsob je v rámci konfiguračního režimu rozhraní, na kterém je zároveň dána síť, která se příkazem zařadí do směrovacího procesu.

```
R4-CE(config)#interface fastEthernet0/0
R4-CE(config-if)#ipv6 ospf 1 area 0
```

Nakonfigurujte směrovací protokol na příslušných směrovačích a přidejte IPv6 sítě včetně loopback rozhraní do smerovacího procesu OSPFv3.

d) Správnost vaší konfigurace si můžete ověřit například příkazem **show ipv6 route**. Na směrovačích R1–PE a R3–PE byste měli vidět následující výpisy.

```
R1-PE#show ipv6 route
<výstup zkrácen>
C   3000:A::/64 [0/0]
     via FastEthernet0/1, directly connected
L   3000:A::2/128 [0/0]
     via FastEthernet0/1, receive
O 3000:ABC::1/128 [110/1]
     via FE80::C804:6FF:FE18:8, FastEthernet0/1
<výstup zkrácen>
```

```
R3-PE#show ipv6 route
<výstup zkrácen>
C   3000:B::/64 [0/0]
     via FastEthernet0/1, directly connected
L   3000:B::2/128 [0/0]
     via FastEthernet0/1, receive
O 3000:AB::1/128 [110/1]
     via FE80::C805:FFF:FEB4:8, FastEthernet0/1
<výstup zkrácen>
```

Dodatečně můžete zkontrolovat konektivitu mezi R1–PE a R4–CE loopback6 a stejně R3–PE a R5–CE loopback6 příkazem **ping**.

e) V tomto kroku přistoupíme ke konfigurací samotného tunelu 6to4. Na směrovači R4–CE vytvoříme rozhraní tunelu a přidělíme mu IPv6 adresu.

```
R4-CE(config)#interface tunnel0
R4-CE(config-if)#ipv6 address 2002:AC10:101::/48
```

První kvartál adresy nese hodnotu 2002 typickou pro 6to4 tunelové adresy. Druhý a třetí kvartál adresy tvoří hexadecimální forma IPv4 adresy rozhraní fastEthernet0/0, která zároveň reprezentuje zdroj tunelu.

f) Dále nastavíme zdroj tunelu jako IPv4 adresu rozhraní fastEthernet0/0 a nastavíme režim tunelu na 6to4.

```
R4-CE(config-if)#tunnel source 172.16.1.1
R4-CE(config-if)#tunnel mode ipv6ip 6to4
```

Klíčový vyraz **ipv6ip** v tomto případě znamená, že se zapouzdřuje IPv6 paket do paketu IPv4. Část výrazu **ip** značí IPv4.

g) Dynamický tunel 6to4 nedokáže pracovat s IGP protokoly a proto musíme nakonfigrovať statickou cestu pro síť 3000:AB::.

```
R4-CE(config)#ipv6 route 3000:AB::/32 2002:AC10:201::
```

Kde adresa 2002:AC10:201:: figuruje jako druhá strana tunelu resp. next hop. Další statickou cestu nakonfigurujeme jako cestu do samotného tunelu.

```
R4-CE(config)#ipv6 route 2002:AC10:201::/48 Tunnel0
```

h) Směrovač R5–CE je nutné nakonfigurovat obdobně jako R4–CE. Vytvoříme rozhraní tunelu a přiřadíme mu IPv6 adresu.

```
R5-CE(config)#interface tunnel0
R5-CE(config-if)#ipv6 address 2002:AC10:201::/48
```

I v tomto případě je druhý a třetí kvartál adresy tvoření hexadecimální hodnotou IPv4 adresy rozhraní fastEthernet0/0 směrovače R5–CE.

i) Obdobně jako tomu bylo u směrovače R4–CE tak i zde nakonfigurujeme zbylé parametry jako zdroj a režim tunelu.

```
R5-CE(config-if)#tunnel source 172.16.2.1
R5-CE(config-if)#tunnel mode ipv6ip 6to4
```

j) Zbývá nám nakonfigurovat statické cesty.

```
R5-CE(config)#ipv6 route 3000:ABC::/32 2002:AC10:201::
R5-CE(config)#ipv6 route 2002:AC10:101::/48 Tunnel0
```

k) Na směrovači R4–CE si zobrazíme informace o nakonfigurovaném tunely příkazem **show interfaces tunnel** *0*. Z níže uvedeného výpisu můžeme vyčíst informace jako IP adresa zdroji tunelu, režim zapouzdření a hodnotu MTU.

```
R4-CE#show interfaces tunnel 0
Tunnel0 is up, line protocol is up
  Hardware is Tunnel
  MTU 17920 bytes, BW 100 Kbit/sec, DLY 50000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation TUNNEL, loopback not set
  Keepalive not set
  Tunnel source 172.16.1.1
  Tunnel protocol/transport IPv6 6to4
  Tunnel TTL 255
  Tunnel transport MTU 1480 bytes
<výstup zkrácen>
```

Příkazem **show ipv6 interface tunnel** *0* si zobrazíme IPv6 informace o konkrétním tunely na směrovači R4–CE.

```
R4-CE#show ipv6 interface tunnel 0
Tunnel0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::AC10:101
  No Virtual link-local address(es):
  Global unicast address(es):
    2002:AC10:101::, subnet is 2002:AC10:101::/48
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF00:0
    FF02::1:FF10:101
  MTU is 1480 bytes
<výstup zkrácen>
```

l) Na směrovači R1–PE si spusťte zachytávání provozu na rozhraní Serial1/0. Ze směrovače R4–CE se příkazem **ping** budeme dotazovat na adresu 3000:AB::1, přičemž zdroj dotazu bude samotné rozhraní loopback6.

```
R4-CE#ping 3000:AB::1 source loopback 6
Sending 5, 100-byte ICMP Echos to 3000:AB::1, timeout is 2 seconds:
Packet sent with a~source address of 3000:ABC::1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 40/53/64 ms
```

Obrázek A.44 ilustruje dotaz ping a jeho zapouzdření v rámci tunelu 6to4. IPv6 paket je zapouzdřený do IPv4 paketu. Oblast MPLS se řídí pouze informacemi v hlavičkách MPLS a IPv4. Dále z ilustrace vidíme, že hlavička IPv4 paketu nese v poli *Protocol* údaj IPv6 s číslem 41. To značí, že protokol vyšší vrstvy je IPv6. Zdrojová a cílová adresa IPv4 paketu je adresa konkrétního zdroje tunelu. V IPv6 hlavičce v poli *Next header* je informace o přenášeném protokolu ICMPv6 s číselným údajem 58. Hlavička MPLS nese *Label* s hodnotou 17, kterou jí přidělil směrovač R1–PE. Hodnota *Bottom of Label Stack* je rovna 1. To značí, že se jedná o poslední a v tomto případě jediný štítek.

```
Source              Destination         Protocol  Length  Info
3000:abc::1         3000:ab::1          ICMPv6    128     Echo (ping) request id=0x1de2, seq=0, hop limit=64
⊞ Frame 16: 128 bytes on wire (1024 bits), 128 bytes captured (1024 bits) on interface 0
⊞ Cisco HDLC
⊟ MultiProtocol Label Switching Header, Label: 17, Exp: 0, S: 1, TTL: 254
    0000 0000 0000 0001 0001 .... .... .... = MPLS Label: 17
    .... .... .... .... .... 000. .... .... = MPLS Experimental Bits: 0
    .... .... .... .... .... ...1 .... .... = MPLS Bottom Of Label Stack: 1
    .... .... .... .... .... .... 1111 1110 = MPLS TTL: 254
⊟ Internet Protocol Version 4, Src: 172.16.1.1 (172.16.1.1), Dst: 172.16.2.1 (172.16.2.1)
    Version: 4
    Header Length: 20 bytes
  ⊞ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT
    Total Length: 120
    Identification: 0x0005 (5)
  ⊞ Flags: 0x00
    Fragment offset: 0
    Time to live: 254
    Protocol: IPv6 (41)
  ⊞ Header checksum: 0x6135 [validation disabled]
    Source: 172.16.1.1 (172.16.1.1)
    Destination: 172.16.2.1 (172.16.2.1)
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
⊟ Internet Protocol Version 6, Src: 3000:abc::1 (3000:abc::1), Dst: 3000:ab::1 (3000:ab::1)
  ⊞ 0110 .... = Version: 6
  ⊞ .... 0000 0000 .... .... .... .... .... = Traffic class: 0x00000000
    .... .... .... 0000 0000 0000 0000 0000 = Flowlabel: 0x00000000
    Payload length: 60
    Next header: ICMPv6 (58)
    Hop limit: 64
    Source: 3000:abc::1 (3000:abc::1)
    Destination: 3000:ab::1 (3000:ab::1)
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
⊞ Internet Control Message Protocol v6
```

Obr. A.44: Dotaz příkazu ping, R1–PE rozhraní Serial1/0 (6to4 režim).

A.4.8 Úkol 5 – MPLS a IPv6 pomocí techniky 6PE

a) Na směrovačích R1–PE a R3–PE vytvoříme loopback rozhraní. Tato rozhraní budou sloužit jako *Router ID* v rámci směrovacího protokolu BGP. Zároveň rozhraní zařadíme do směrovacího procesu OSPFv2.

```
R1-PE(config)#interface loopback0
R1-PE(config-if)#ip address 1.1.1.1 255.255.255.255
R1-PE(config-if)#ip ospf 1 area 0

R3-PE(config)#interface loopback0
R3-PE(config-if)#ip address 2.2.2.2 255.255.255.255
R3-PE(config-if)#ip ospf 1 area 0
```

b) Dále spustíme směrovací protokol BGP s autonomním číslem 65 000. Přiřadíme mu *Router ID* podle loopback rozhraní a nakonfigurujeme sousedství.

```
R1-PE(config)#router bgp 65000
R1-PE(config-router)#bgp router-id 1.1.1.1
R1-PE(config-router)#neighbor 2.2.2.2 remote-as 65000
R1-PE(config-router)#neighbor 2.2.2.2 update-source loopback0

R3-PE(config)#router bgp 65000
R3-PE(config-router)#bgp router-id 2.2.2.2
R3-PE(config-router)#neighbor 1.1.1.1 remote-as 65000
R3-PE(config-router)#neighbor 1.1.1.1 update-source loopback0
```

c) Příkazem **show ip bgp summary** ověřte navázání sousedství iBGP v rámci směrovačů R1–PE a R3–PE.

```
R1-PE#show ip bgp summary
BGP router identifier 1.1.1.1, local AS number 65000
BGP table version is 1, main routing table version 1

Neighbor        V AS   MsgRcvd  MsgSent  TblVer  InQ OutQ  Up/Down
2.2.2.2         4   65000       116      116       1    0    0  00:00:23
```

d) Na směrovači R1–PE v režimu směrovacího procesu BGP povolíme IPv6 funkci a tím zprovozníme MP–BGP. V rámci tohoto režimu nakonfigurujeme jen základní nastavení jako sousedství mezi R1–PE a R3–PE, zasílání MPLS štítků danému sousedova a redistribuci OSPFv3 cest. Metriku redistribuovaných cest nastavíme na hodnotu 5.

```
R1-PE(config-router)#address-family ipv6
R1-PE(config-router-af)#neighbor 2.2.2.2 activate
R1-PE(config-router-af)#neighbor 2.2.2.2 send-label
R1-PE(config-router-af)#redistribute ospf 1 metric 5
R1-PE(config-router-af)#exit-address-family
```

Na směrovači R3–PE samostatně proveďte obdobné nastavení. Samozřejmě v případě R3–PE bude souseda reprezentovat *Router ID 1.1.1.1*.

e) Zbývá nám poslední krok a tím je redistribuovat cesty z BGP do směrovacího procesu OSPFv3. V tomto případě nastavíme metriku redistribuovaných cest na hodnotu 3.

```
R1-PE(config)#ipv6 router ospf 1
R1-PE(config-router)#redistribute bgp 65000 metric 3
```

Stejné kroky proveďte ina směrovači R3–PE.

f) Na směrovači R4–CE si zobrazíme stav směrovací tabulky příkazem **show ipv6 route**.

```
R4-CE#show ipv6 route
<výstup zkrácen>
C   3000:A::/64 [0/0]
     via FastEthernet0/0, directly connected
L   3000:A::1/128 [0/0]
     via FastEthernet0/0, receive
OE2 3000:AB::1/128 [110/3]
     via FE80::C801:1FFF:FE0C:6, FastEthernet0/0
<výstup zkrácen>
```

Z výpisu vidíme, že cesta do sítě *3000:AB::1/128* je označena jako externí cesta typu 2. Dále si zobrazte směrovací tabulku směrovače R1–PE příkazem **show ipv6 route**.

```
R1-PE#show ipv6 route
<výstup zkrácen>
C   3000:A::/64 [0/0]
     via FastEthernet0/1, directly connected
L   3000:A::2/128 [0/0]
     via FastEthernet0/1, receive
B   3000:AB::1/128 [200/5]
     via 2.2.2.2%default, indirectly connected
<výstup zkrácen>
```

Můžeme vidět údaj, který se váže k síti *3000:AB::1/128* jako *2.2.2.2%default*. Zobrazíme si tabulku na směrovači R1–PE, která reprezentuje informace o IPv6 sítích BGP příkazem **show bgp ipv6 unicast**.

```
R1-PE#show bgp ipv6 unicast
<výstup zkrácen>
     Network          Next Hop            Metric LocPrf Weight Path
 *>i 3000:AB::1/128   ::FFFF:2.2.2.2           5    100      0 ?
 *>  3000:ABC::1/128  FE80::C804:6FF:FE18:8
                                                5         32768 ?
```

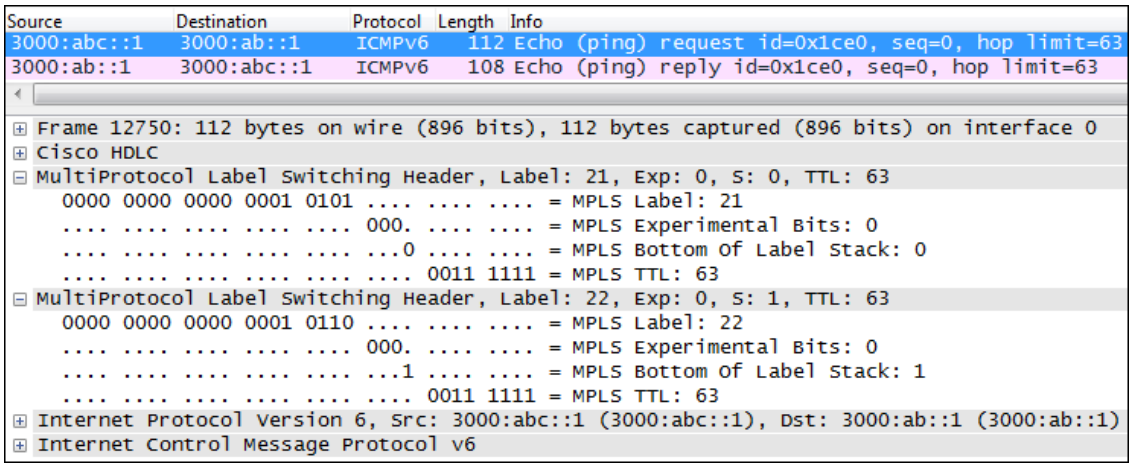
Z výše uvedeného výpisu vidíme, že adresa loopback rozhraní na směrovači R3–PE *2.2.2.2* se namapovala do IPv6 adresy, kterou směrovač R1–PE používá jako next hop pro síť *3000:AB::1/128*. Dále si příkazem **show mpls forwarding–table** na směrovači R1–PE zobrazíme LFIB tabulku. Z výpisu uvedeného níže vidíme, že k síti *2.2.2.2* je přiřazena hodnota štítku 21. Hodnota 21 se používá k prvnímu štítku, přičemž slouží k dosažení směrovače R3–PE. Hodnota 22 se použije pro druhý štítek, přičemž poukazuje, že se jedná o IPv6 paket.

```
R1-PE#show mpls forwarding-table
Local Outgoing  Prefix          Bytes Label Outgoing  Next Hop
Label Label     or Tunnel Id    Switched    interface
16    Pop Label 50.50.50.50/32  0           Se1/0     point2point
17    Pop Label 192.168.2.0/24  0           Se1/0     point2point
18    16        172.16.2.0/24   0           Se1/0     point2point
19    No Label  10.1.1.1/32     0           Fa0/1     172.16.1.1
20    18        10.1.2.1/32     0           Se1/0     point2point
21    21        2.2.2.2/32      0           Se1/0     point2point
22    No Label  3000:ABC::1/128 1140        Fa0/1     FE80::C804:6FF:FE18:8
```

g) Spusťte si zachytávání provozu na směrovači R1–PE rozhraní Serial1/0. Ze směrovače R4–CE se budeme dotazovat příkazem **ping** na adresu *3000:AB::1*, přičemž použijeme loopback6 jako zdroj dotazu.

```
R4-CE#ping 3000:AB::1 source loopback6
```

Z obrázku A.45 vidíme praktický důkaz toho, že v rámci sítě MPLS se IPv6 komunikace bude řídit dvěma štítky. Jak bylo uvedeno v předchozím kroku, hodnota 21 se využije jako parametr pro dosažení směrovače R3–PE. Hodnota štítku 22 poukazuje na paket IPv6. V rámci této části úkolu si je třeba uvědomit, že směrovače typu P, čili jádro MPLS, se v rámci přepínání paketů řídí pouze obsahem štítku nehledě na to, že se přenáší paket IPv6.



Obr. A.45: Dotaz příkazu ping, R1–PE rozhraní Serial1/0 (6PE režim).

A.4.9 Kontrolní otázky

1. Na jakém principu pracuje protokol MPLS a jaká je jeho výhoda oproti klasickému IP směřování?
2. Jaké protokoly znáte, jejichž úkolem je distribuce štítků MPLS?
3. Na základě jakého parametru jsou směrovače v rámci MPLS oblasti jednoznačně identifikovány?
4. Na jaké vrstvě síťového modelu je řešena autentizace MPLS a jaký algoritmus se k tomu využívá?
5. V čem spočívá myšlenka 6to4 tunelu v rámci propojení IPv6 sítí, které jsou oddělené MPLS oblastí?
6. V čem spočívá myšlenka 6PE v rámci propojení IPv6 sítí, které jsou oddělené MPLS oblastí?

A.4.10 Samostatná úloha

Samostatně nakonfigurujte v rámci celé MPLS sítě autentizaci LDP zpráv.

Poznámka: Přiložené DVD obsahuje konfiguraci zařízení před a po samostatné úloze. Dále obsahuje samotné zdrojové soubory projektů pro GNS3.