DELTA – Střední škola informatiky a ekonomie, s. r. o.

Ke kamenci 151, Pardubice

Webový portál pro podporu výuky dynamických směrovacích protokolů EIGRP a OSPF

Jméno a příjmení: Vít Tintěra

Třída: 4. B

Obor: Informační technologie 18-20-M/01

Pro školní rok: 2023/2024

Zadání maturitního projektu z informatických předmětů

Jméno a příjmení:

Pro školní rok: *2023/2024*

Třída: *4.*

Obor: *Informační technologie 18-20-M/01*

Téma práce: Webový portál pro podporu výuky dynamických směrovacích protokolů EIGRP a OSPF

Vedoucí práce: Josef Horálek

Způsob zpracování, cíle práce, pokyny k obsahu a rozsahu práce:
Cílem práce je podrobně popsat principy fungování dynamických směrovacích protokolů EIGRP a OSPF, porovnat jejich vlastnosti a chování a vytvořit webový portál pro podporu rozšiřujícího studia dynamického směrování.

Stručný časový harmonogram (s daty a konkretizovanými úkoly):
Dynamické směrovací protokoly

Úvod do protokolu EIGRP a OSPF

Princip výměny informací směrovači s protokoly EIGRP a OSPF

Konfigurace protokolu EIGRP a OSPF

Řešení problémů spojených s fungováním protokolů EIGRP a OSPF

Návrh laboratorních úloh pro ověření znalostí protokolů EIGRP a OSPF

Vytvoření webového portálu

Prohlašuji, že jsem maturitní projekt vypracoval samostatně, výhradně s použitím uvedené literatury.

V Rábech dne 2. 4. 2024

(vlastnoruční podpis)

Děkuji vedoucímu doc. Mgr. Josefu Horálkovi, Ph.D. za odborné vedení, studijní materiály, podporu a trpělivost při zpracovávání maturitního projektu.

Klíčová slova

Směrování v sítích	Routing in networks
Dynamické směrování	Dynamic routing
Směrovací protokoly	Routing Protocols
Konvergence	Convergence
Algoritmus	Algorithm
Konfigurace	Configuration
Laboratorní úlohy	Practical exercises
Učební podklady	Teaching materials

Resumé

Tato práce obsahuje materiály pro porozumění tématu dynamického směrování, vhodné jak do výuky tak pro samostudium. Přiloženy jsou také praktické úlohy pro vyzkoušení a upevnění si nově nabytých znalostí.

Obsah

Klíčová slova 5
Resumé 5
1. ÚVOD 8
1.1. Internet a směrování informací 8
1.2. Interní směrovací protokoly 9
2. Distance-vector protokoly 9
2.1.1. Routing Information Protocol 10
2.1.1.1. RIPv1 11
2.1.1.2. RIPv2 12
2.1.1.3. RIPng 12
3. Link-state protokoly 12
3.1. OSPF 13
3.1.1. Oblasti 13
3.1.1.1. Směrování a oblasti 13
3.1.2. Směrování v různých sítích 14
3.1.3. Link State Advertisement (LSA) 14
3.1.3.1. Typy LSA 14
3.1.3.2. Hlavička LSA 15
3.1.3.3. Zpracování LSA 15
3.1.4. Přenos dat v OSPF 15
3.1.5. Typy paketů OSPF 16
3.1.5.1. Hello paket (typ 1) 16
3.1.5.2. Database description (typ 2) 16
3.1.5.3. Link state request (typ 3) 16
3.1.5.4. Link state update (typ 4) 16
3.1.5.5. Link-state acknowledgment (typ 5) 16
3.1.6. Výpočet cesty 17
4. EIGRP – “Hybridní“ směrovací protokol 17
4.1. Algoritmus DUAL 17

4.2. Moduly a RTP....17
4.3. Data a jejich přenos v rámci EIGRP....18
4.3.1. Tabulky....18
4.3.1.1. Tabulka sousedů....18
4.3.1.2. Směrovací tabulka....18
4.3.1.3. Tabulka síťové topologie....18
4.3.2. Pakety....19
4.3.2.1. Kontaktní Hello pakety....19
4.3.2.2. Potvrzovací pakety....19
4.3.2.3. Aktualizační pakety....19
4.3.2.4. Paket dotazu a odpovědi....19

1. ÚVOD

1.1. Internet a směrování informací

Základní jednotkou sítě jsou koncová zařízení – ta, která data buď přijímají, nebo odesílají. Pokud nejsou tato zařízení přímo spojena, musí přenášená informace projít přes síť.

Internet je celosvětová komunikační síť. Skládá se z mnoha jednotlivých sítí, komunikujících mezi sebou, a ty se zase skládají z mnoha jednotlivých zařízení, které spolu také komunikují. Tato komunikace by nebyla možná bez směrování.

Směrování je totiž hledání cesty v počítačové síti od odesílajícího zařízení až pro příjemce. Tento úkol mnohdy není zrovna jednoduchý, a proto ho plní armády specializovaných zařízení – směrovačů neboli anglicky routerů. Ty mají své předem dané instrukce, jak informace směrovat, tak aby se dostaly do cíle.

Tyto instrukce mohou být předem ručně nastavené a neměnné, a tento způsob nazýváme **statické směrování**. Jeho hlavní výhodou je, že přesně víme, kudy informace putují, a systém je bezpečný. Ovšem tento přístup má fatální chybu, neumí se totiž přizpůsobit změnám v topologii sítě, například výpadku spojovací linky nebo směrovače. Proto se tento způsob využívá jen v těch nejmenších sítích.

Oproti tomu **dynamické směrování** přináší mnoho výhod. Síť se automaticky přizpůsobuje změnám v topologii, informace putují po nejlepší vypočítané cestě a může se počítat i s kapacitou jednotlivých cest a informace dělit a posílat paralelně. Konkrétní chování autonomního systému s dynamickým směrování záleží zcela na zvoleném **směrovacím protokolu**.

Směrovací protokol je sada instrukcí, která definuje formát, ve kterém si směrovače mezi sebou vyměňují informace. Podle těchto informací se postupně učí topologii sítě. Důležitou součástí směrovacího protokolu je také algoritmus, který na základě předem zmíněných informací vypočítá cestu, kterou pak data poputují.

Propojené směrovače se stejným směrovacím protokolem tvoří druh sítě, který nazýváme **autonomní systém**.

Směrovací protokoly se dělí na dvě skupiny, interní a externí. Interní jsou schopny uložit si do paměti celou strukturu sítě a podle toho se pak rozhodovat. Kvůli limitacím velikosti paměti jsou schopny směrovat data pouze uvnitř autonomního systému, který nepřekračuje určitý rozměr.

Oproti tomu externí směrovací protokoly směrují data mezi různými autonomními systémy. Směrují data na okraj autonomního systému, kde je předají “hraničnímu” směrovači, a vnitřní směrovací protokol už najde cestu ke konkrétnímu cílovému zařízení. K tomuto druhu

směrovacích protokolů patří již dnes zastaralý Exterior Gateway Protocol a současně používaný Border Gateway Protocol.

Cílem této práce je popsat různé způsoby směrování uvnitř autonomního systému a předat znalosti základní konfigurace několika protokolů.

1.2. Interní směrovací protokoly

Vnitřní směrovací protokoly směrují data uvnitř autonomního systému. Jeden autonomní systém může využít i více protokolů najednou, nezávisle na sobě. Směrovače vždy předávají data nejbližšímu sousedovi na cestě k cíli. Dělíme je do dvou skupin, podle toho, který algoritmus používají na vypočítání nejlepší cesty do cíle.

2. Distance-vector protokoly

Jak už název napovídá, pro tuto skupinu protokolů jsou důležité dvě věci: vzdálenost do cíle, a směr k cíli. Vzdálenost se typicky určuje v počtech „přeskoků“, tedy přes kolik směrovačů musí data přejít, než se dostanou k cíli. Směr určuje, přes který ze sousedních směrovačů vede nejkratší cesta do cíle. Výhodou tohoto přístupu je, že směrovač vůbec nemusí znát topologii sítě.

Tyto informace se šíří tím, že každý směrovač periodicky zasílá celou svoji směrovací tabulku všem svým sousedům. Na začátku směrovací tabulka každého směrovače obsahuje pouze přímo připojená zařízení, ovšem postupně se rozšiřuje o přicházející tabulky sousedů, a tudíž i o zařízení k nim připojená. U každého záznamu je uvedeno, kolik přeskoků musel překonat, než se dostal do současné tabulky. Tato upravená tabulka se pak znovu rozešle všem sousedům.

Takto se informace o zařízeních a jejich relativní poloze šíří celou sítí, než mají všechny směrovače informace o všech zařízeních a cestám k nim, a tím nastane **konvergence**.

Je očividné, že tento způsob má několik nedostatků. Jedním je neustálé zasílání mírně pozměněných směrovacích tabulek všemi směrovači všem svým sousedům, což ve větších sítích způsobuje značnou zátěž přenosních kapacit. Dalším je dlouhá doba, než nastane konvergence, protože šíření informací o změnách v síti probíhá poměrně pomalu. To způsobuje problém při jakékoliv změně v topologii, sytém je tedy poměrně neflexibilní. Navíc při výpadku spojení nebo směrovače není nikdo, kdo by šířil, že daná cesta už neexistuje. Čeká se prostě daný čas, a když nepřijde update této cesty, tak se považuje za nefunkční.

Problémem je i samotná volba cesty. Tento systém počítá sice počet přeskoků, a tím i přibližnou vzdálenost, ale vůbec nezohledňuje rychlost a kapacitu jednotlivých linek, což vede v některých případech k volbě suboptimální cesty.

Aby nebyla zmíněna pouze negativa, tak tato skupina protokolů má i značnou výhodu, a to je v jednoduchosti implementace a nízkých nároků na výpočetní výkon jednotlivých směrovačů. To je také proto, že nejstarší z těchto protokolů byly používány na samém počátku dynamického

směrování, kdy výkon směrovačů byl nízký a sítě poměrně malé. V podobných situacích se používají distance-vector protokoly i v dnešní době.

U tohoto systému šíření informací ovšem nastává jeden zásadní problém. Pokud vypadne jeden ze směrovačů (označme ho třeba X), jeho soused (směrovač A) na to po chvíli přijde, a podívá se do své směrovací tabulky, jestli neexistují nějaké alternativní cesty, jak se k vypadlému směrovači dostat. Tady se může stát, že v tabulce má přejatý záznam od svého souseda (směrovače C), který ještě nezjistil, že jeho cesta k směrovači X nefunguje, a chybně ji propaguje jako funkční. Tato cesta je v našem příkladě 2 skoky dlouhá, a vede přes C-A-X. Směrovač A tuto cestu převezme, aniž by si uvědomil, že cesta vede přes něj samého, protože má informaci jenom o počtu přeskoků, ale ne už, kudy přesně cesta vede. Vytvoří tedy cestu A-C-A-X, 3 přeskoky dlouhou, přidá si ji do směrovací tabulky, a tu rozešle všem sousedům. Mezitím směrovač C přijde na to, že jeho dosavadní cesta (C-A-X) nefunguje, a podívá se, jestli nemá v tabulce alternativní cestu. Jenže to už stihla dorazit od směrovače A chybná cesta (A-C-A-X), a směrovač C ji převzal. C si tedy vytvoří cestu C-A-C-A-X, 4 přeskoky dlouhou, a přepošle ji spolu se svojí směrovací tabulkou všem sousedům. Jelikož směrovač A mezitím zjistil, že jeho cesta je nefunkční, a hledá alternativu, celý proces může začít znovu… Aniž by někomu v síti došlo, že směrovač X prostě nefunguje.

Je několik způsobů, jak tento problém řešit. Jedním z nich je použití tzv. **Split horizon**. V praxi to znamená, že směrovač neposílá informace o cestě tomu sousedovi, od kterého tuto cestu přebral. Tím se předejde výše zmíněnému problému. Další způsob, značně jednodušší, je stanovit horní hranici počtu přeskoků (např. protokol RIP ji má nastavenou na 15), a pokud je přesažena, brát tuto cestu jako nefunkční. To znamená, že popisovaný problém sice nastane, ale po několika opakováních směrovače A a C dojdou k správnému závěru, že směrovač X je nedosažitelný.

2.1.1. Routing Information Protocol

Routing Information Protocol, zkráceně RIP, je jeden z nejstarších dynamických směrovacích protokolů.

RIP používá jako transportní protokol User Datagram Protocol (UDP). Na měření cesty využívá již zmíněný počet přeskoků, implementuje i split horizon. Na prevenci směrovacích smyček používá i několik dalších vylepšení: **Poison reverse**, která funguje podobně jako split horizon, s tou výjimkou, že místo nezasílání informací směrem ke zdroji, tak se zasílá cesta s hodnotou nekonečno, aby se tak urychlilo rozpoznání linky jako nefunkční v případě výpadku. **Hold down**, která v případě ztráty nejlepší cesty po určený čas nepřijímá updaty o této cestě, protože by mohlo jít o chybné záznamy od směrovačů, které si zatím výpadku nevšimly. Nevýhodou je opět zpomalení konvergence.

Distance-vector protokoly začaly být implementovány zhruba od roku 1969 společně s rozvojem veřejných sítí jako ARPANET a CYCLADES. Nejstarší předchůdce RIP je GWINFO (Gateway Information Protocol), směrovací protokol vytvořený Xeroxem pro svou experimentální síť. Ten byl vylepšen na XNS RIP (Xerox Network Systems Routing Information Protocol). Z tohoto základu vzniklo mnoho podobných protokolů: IPX RIP, RTMP, IP RIP a další. Implementován byl i v systému UNIX, verzi 4.2BSD, která se stala velmi populární. Z toho vyplývá, že princip protokolu RIP byl využíván hojně již před standardizací provedenou v roce 1988, sjednotila různé protokoly do verze RIPv1.

2.1.1.1. RIPv1

První standart protokolu RIP byl sepsán Charlesem Hedrickem v roce 1988. Po zapnutí směrovač posílá na všechna připojená zařízení (tomuto způsobu komunikace říkáme **broadcast**) požadavek, aby mu byla zaslána směrovací tabulka. Okolní směrovače na tuto zprávu odpoví zasláním své směrovací tabulky. Směrovač, který posílal dotaz si pak upraví svoji tabulku podle nových informací. V směrovací tabulce je uložen pro každý cíl vždy jeden záznam, ten s nejkratším počtem přeskoků. Pokud se vyskytne další cesta do stejného cíle se stejným počtem přeskoků, je přidána jako další záznam. Tyto cesty směrovač využije k rozprostření informací na více spojení.

Požadavek na zaslání tabulky je rozesílán každých 30 sekund. Tento čas je u každého směrovače náhodně pozměněn, aby nedošlo k synchronizaci broadcastů mezi několika směrovači a tím k zbytečným výkyvům v zátěži sítě. Některé směrovače také můžou být nastaveny tak, že jiné tabulky přijímají, ale vlastní tabulku neodesílají. V tom případě je nazýváme „tichými“ směrovači.

Je důležité poznamenat, že RIPv1 *nepodporuje* VLSM(variable length subnet masks). Používá se zde starší směrování pomocí tříd. To znamená, že nemohou být tvořeny subnety různých délek, síť je rozdělená na několik „tříd“, skládajících se vždy z adresy sítě a adres zařízení.

- Třída A: První oktet je adresa sítě, zbylé tři jsou pro zařízení.
    - Rozsah: 1.0.0.0 až 126.0.0.0
- Třída B: První dva oktety jsou adresa sítě, a zbylé dva adresy zařízení.
    - Rozsah: 128.0.0.0 až 191.255.0.0
- Třída C: První tři oktety pro síť, jeden pro zařízení.
    - Rozsah: 192.0.0.0 až 223.255.255.0

Tento způsob dělení adres se již dnes nepoužívá, a z dobrého důvodu. Nejdou zde tvořit různě velké subnety podle potřeby, musí být vždy velikostí spadat do jedné ze zmíněných tříd. To je sice o něco jednodušší než dnes používané způsoby, ale přináší to s sebou značnou neefektivitu při přidělování adres, což vede k menšímu počtu použitelných adres v síti.

RIPv1 také nepodporuje používání autentizace směrovačů, což znamená, že se zbytečně vystavuje případným útokům.

2.1.1.2. RIPv2

RIPv2 vznikl v roce 1944, kdy již existovaly ostatní, mnohem pokročilejší směrovací algoritmy, jako třeba OSPF nebo IS-IS. Ovšem díky nízkým nárokům na nasazení a údržbu, a nízkou zátěží přenosních kapacit v malých sítích, byl RIP stále mnohem populárnější než jeho nová konkurence. Proto bylo potřeba stárnoucí RIP vylepšit, aby mohl pokračovat ve službě.

Nejdůležitější ze změn bylo implementování VLSM a podpora CIDR (Classless Inter-Domain Routing). Tím se zrušilo předchozí dělení adres na třídy a přešlo se na plnou podporu subnetů tak jak je známe dnes. Další bylo přejití z broadcastu směrovacích tabulek všem zařízením okolo na pouze multicast okolním směrovačům, což pomáhá snížit zátěž sítě. Nová byla také autentizace komunikace mezi směrovači, nejdříve ověřovaná pouze nezašifrovaným heslem, později v roce 1997 přešla na kryptografický algoritmus MD5. Poslední vylepšení je tzv. „route tag“ u cest ve směrovací tabulce, umožňující zjistit, jestli byla daná cesta převzata od jiného protokolu nebo sítě, či nikoliv.

RIPv2 také obsahuje možnost plné zpětné kompatibility s protokolem RIP.

2.1.1.3. RIPng

Při vývoji bylo cílem modifikovat RIPv2 co nejméně tak, aby byl schopen používat nový standard IPv6. Výsledkem tohoto snažení je RIPng, který je velmi podobný RIPv2, s několika zásadními změnami.

Místo pro adresu bylo změněno z 32 bitů na 128, aby bylo možno přenášet IPv6 adresy. Maska podsítě byla změněna na prefix používaný v IPv6. Byla přidána možnost pro odesílatele updatu specifikovat jiný směrovač než sám sebe jako směr pro určitou cestu, což vede k mírné optimizaci směrování ve specifických případech. Autentizace zpráv byla úplně zrušena, protože IPv6 má vlastní zabudovanou autentizaci IPsec. Byla zrušena limitace na velikost packetu. Maximální počet přeskoků zůstal na 15, protože to bylo uznáno za dobrý kompromis mezi maximální velikostí sítě a časem pro odstranění nastalých směrovacích smyček.

Protože změn je velmi málo, RIPng má zhruba stejné použití jako RIPv2, totiž jako jednoduchý protokol pro malé až střední sítě s nižší komplexitou topologie.

3. Link-state protokoly

Link-state protokoly směrují data na základě znalosti stavu jednotlivých linek (spojů) v síti. Stavem se myslí, jestli jsou funkční, a jejich „cena“, obvykle složená z rychlosti, kapacity, nebo může být manuálně nastavená. Každý směrovač v síti zná celou topologii sítě a ceny jednotlivých spojení (v značném kontrastu s distance-vector algoritmy, kdy směrovač zná jenom cíle a směr k nim). Směrovače mají uloženou topologii v tzv. topologické databázi, a všechny směrovače

mají uloženou stejnou topologii. Topologie je uložena v podobě souvislého ohodnoceného grafu, kde vrcholy reprezentují směrovače a hrany jednotlivá spojení. Každý směrovač si pomocí Dijkstrova algoritmu počítá nejlepší cestu ke všem ostatním směrovačům, a z toho staví směrovací tabulku. Každý směrovač také sleduje funkčnost a cenu přímo připojených linek, a pokud u některé nastane změna, okamžitě rozšíří tuto informaci všem ostatním směrovačům v síti, a ti si uloží tento nový stav do topologické databáze. Díky této pohotové reakci na jakoukoliv změnu v topologii síti je dosaženo rychlé konvergence při výpadu či náběhu linky.

Jak je již patrné, nedochází zde k žádnému periodickému zasílání směrovacích tabulek, rozesílají se pouze změny. Pokud je nastartován nový směrovač v již zaběhnuté síti, místo postupného budování topologie sítě si ji prostě může vyžádat od sousedního směrovače.

3.1. OSPF

Protokol OSPF byl vytvořen v letech 1988–1991 organizací IETF (Internet Engineering Task Force). Je to dynamický interní (vnitřní) směrovací protokol (IGP) a řadí se k link-state protokolům. To znamená, že si buduje plnou topologickou databázi sítě včetně stavu a ceny všech spojů. Tato data jsou v topologické databázi uložena v podobě stromu, kde směrovač je kořenem a linky jsou větve. Používá algoritmus SPF (shortest path first), což je modifikace Dijkstrova algoritmu hledání cestu v grafu. Plně podporuje CIDR (vytváření různě velkých subnetů) a označování převzatých informací z cizích protokolů. Používá IP multicast na zasílání aktualizací směrovacích tabulek, a podporuje bezpečnostní kontrolu aktualizací. To vše přispívá k tomu, že protokol OSPF je rychlejší při detekci změn a přizpůsobení se jim, ale zároveň zatěžuje síť méně než starší protokoly.

OSPF směruje pakety pouze podle IP adresy obsažené v nich. Nepřidává k nim žádné další informace v žádné fázi cesty. Je přizpůsoben přímo na směrování protokolu IP. Je sice složitětější na implementaci a údržbu, ale tato nevýhoda je bohatě vyvážena robustností a šířkou nabízených funkcí.

3.1.1. Oblasti

Sítě OSPF se dělí na menší oblasti, a to má dva důvody: lepší škálovatelnost a rychlejší konvergence. Škálovatelnost proto, že je lepší přidat další oblast než rozšiřovat základní síť. Konvergence je také mnohem rychlejší a stabilnější, když se děje po menších oblastech. Oblasti jsou definovány číslem nastaveným ve směrovači. Základní oblastí je oblast 0.

3.1.1.1. Směrování a oblasti

Známe v 3 typy směrovačů:

- Páteřní směrovač: Pátěřním směrovačem je každý směrovač v oblasti 0.
- Interní směrovač: Směrovač uvnitř oblasti.
- Hraniční směrovač (Border Router, BR): Směrovač, který je připojen ke dvěma nebo více oblastem zároveň. Je zodpovědný za správnou komunikaci mezi oblastmi.

Všechny cesty mezi dvěma různými oblastmi musí procházet přes páteřní oblast (oblast 0), i když jsou dvě oblasti přímo propojené.

3.1.2. Směrování v různých sítích

Protokol OSPF se umí propojit se sítěmi i s jiným směrovacím protokolem, v tomto případě budou cesty takto převzaté označené jako *externí*.

OSPF se umí propojit i s dalšími sítěmi, které také používají protokol OSPF. Proto, aby se sítě vzájemně odlišily, každá má svůj jedinečný identifikátor, kterým je *číslo autonomního systému*. Síti OSPF se tedy říká *Autonomní Systém* (AS).

Zde přijde do hry poslední typ směrovače: ASBR (Autonomous System Border Router). Tento speciální typ směrovače má za úkol zajistit komunikaci mezi sítěmi. Slučuje za tímto účelem informace ze svého AS a odesílá je svému protějšku.

3.1.3. Link State Advertisement (LSA)

OSPF je link-state protokol. To znamená, že je postavený na informacích o stavu linek (spojů) mezi jednotlivými směrovači. Aby si směrovač mohl vytvořit mapu sítě a správně směrovat, potřebuje vědět aktuální stav všech linek v síti. Tyto informace se v síti šíří pomocí zprávy Link State Advertisement. Tu si nyní podrobně rozebereme.

Interní směrovače LSA rozesílají pouze směrovačům ve své oblasti a adresují je přímo recipientům. Důvod k šíření LSA pouze v rámci oblasti je několik: konvergence je rychlejší v menších celcích, a síť je odolnější proti destabilizaci při výpadcích, protože následná konvergence se děje jen v rámci oblasti a neohrožuje zbytek sítě.

Hraniční směrovače se chovají podobně jako interní směrovače, ale separují svoje připojené oblasti. Ještě mají navíc úkol sdružovat adresy v připojených oblastích, a sdílet je s páteřní oblastí (skrz *LSA 3*).

Páteřní směrovače se chovají jako interní, ale protože jsou v páteřní síti, tak se přes ně občas předává informacie o topologii chodící mezi oblastmi.

3.1.3.1. Typy LSA

- Typ 1 – Oznámení LSA o lince směrovače
  - Obsahuje informace o stav linek na cestě od směrovače do jiné oblasti. Interní směrovače mají 1 takovou cestu (která vede k BR), hraniční jich mají víc.
- Typ 2 – Oznámení LSA o síťové lince
  - Obsahuje agregace všech informací o linkách, jejich stavu a nákladech v celé síti. Tyto informace kvůli jejich velikosti a komplikovanosti shromažďuje jen 2 směrovače v celé síti. Je to *Designated Router* (DR) a *Backup Designated Router* (BDR), který nahradí DR v případě výpadku.

- Typ 3 – Souhrnné oznámení LSA s cestou do sítě IP
    - Slouží pro odeslání souhrnu směrovacích informací o oblasti do sousední oblasti. Generuje ji hraniční router (BR).
    - Název je poněkud zavádějící, protože je přežitkem z první verze protokolu OSPF, která používala jiné pojmenování.
- Typ 4 – Souhrnné oznámení LSA s cestou do hraničního směrovače AS
    - Funguje podobně jako typ 3, ale místo mezi oblastmi, tak posílá souhrn mezi celými sítěmi. Generuje ASBR.
- Typ 5 – Oznámení LSA externích linek AS
    - Obsahuje popis externích cílů mimo síť OSPF. Generuje jej hraniční směrovač sousedící s externí sítí.

3.1.3.2. Hlavička LSA

Všechny zmíněné typy LSA mají standartní hlavičku s několika společnými informacemi. To jsou:

- Stáří LSA (v sekundách od vytvoření)
- Nastavení OSPF
- Link State ID
- ID oznamujícího směrovače
- Pořadové číslo LSA (Určuje, jako kolikáté bylo směrovačem vygenerováno. Pomocí něho a stáří LSA se předchází problémům s duplicitou LSA v síti.)
- Kontrolní součet LSA (ověření, jestli LSA není poškozen)
- Délka LSA

Po hlavičce následuje vlastní tělo s daty LSA.

3.1.3.3. Zpracování LSA

Přijaté LSA nemůžou být směrovačem přímo využity k směrování (na rozdíl třeba od zpráv protokolu RIP). Následuje ještě několik kroků: Po příjmu se informace z LSA použijí k aktualizaci topologické databáze (která obsahuje “mapu“ sítě), nad tou se pak spustí algoritmus SPF, a až výsledek jeho výpočtu se porovná se směrovací databází (a pokud je nová cesta lepší, tak se směrovací databáze aktualizuje na tuto novou optimální cestu).

3.1.4. Přenos dat v OSPF

V OSPF existuje 5 typů paketů s jejichž pomocí mezi sebou směrovače komunikují. Nezávisle na typu mají všechny pakety protokolu OSPF společnou hlavičku obsahující následující informace:

- Číslo verze (existují pouze verze 1 a 2, přičemž v1 je velmi zastaralá)
- Typ paketu (viz níže)
- Délka paketu
- ID odesílajícího směrovače

- ID oblasti
- Kontrolní součet (pokud nesouhlasí, paket je poškozen a paket se automaticky zahazuje)
- Typ autentizace (od žádné přes SIMPLE až po MD5, výrazně zvyšuje bezpečnost sítě)
- Tělo autentizace (pokud není platné, paket se zahazuje)

3.1.5. Typy paketů OSPF

3.1.5.1. Hello paket (typ 1)

Hello pakety jsou vyměňovány mezi sousedními směrovači. Slouží pro prvotní navázání vztahu *přilehlosti* (sousedství) a započetí komunikace. Při navázání komunikace se síť musí shodnout na:

- Masce sítě
- Intervalu Hello – jak často probíhá pravidelné ohlášení svého stavu sousedům
- Intervalu Dead – za jak dlouho může být považován směrovač, který se nehlásí za nefunkční

Po úspěšném početí komunikace se pakety Hello používají k tomu, aby si sousedé navzájem dávali najevo, že jsou funkční a schopni komunikace.

Součástí Hello paketu je také seznam sousedů odesílajícího směrovače a také adresa DR a BDR.

3.1.5.2. Database description (typ 2)

Tento typ paketů se používá na přenos popisu databáze. Odesílá se při inicializaci přilehlosti, kdy si sousedé navzájem vyměňují databáze. Později se posílá jako odpověď na žádost.

3.1.5.3. Link state request (typ 3)

Když po přijetí database description směrovač zjistí, že soused má aktuálnější část databáze, odešle sousedovi LSR s přesnými požadavky:

- Číslo typu stavu linky (1-5) (viz *Typy LSA*)
- Identifikátor linky (LS ID)
- ID oznamujícího směrovače

3.1.5.4. Link state update (typ 4)

Generuje se jako odpověď na požadavek (viz předchozí typ 3). Obsahuje LSA *(viz Link State Advertisement)* vyžádaných linek.

3.1.5.5. Link-state acknowledgment (typ 5)

Slouží k potvrzení příjmu LSA (Link-state advertisement). LSA jsou posílány spolehlivě, aby nedošlo k nedorozumění a neúplné konvergenci, proto se pokaždé musí potvrdit jejich příjem. LSAck obsahuje jednoznačnou identifikaci, za které LSA potvrzuje přijetí. Může takto potvrdit i více najednou.

3.1.6. Výpočet cesty

V OSPF má každá linka „náklad“, který určuje, jak výhodná je na použití. Nižší náklad znamená, že OSPF bere tuto linku jako lepší a bude ji upřednostňovat před linkami s vyšším nákladem. Automaticky se náklady vypočtou podle šířky pásma linek. Je možné je manuálně změnit, a může to být použito na jemnější kontrolu toku dat fyzickou sítí, ale vede to k neefektivitám v zatížení, a v drtivé většině případů je automatický systém tou nejlepší volbou.

Seznam všech linek je uložen v topologické databázi v podobě stromu, kde směrovač je kořen a linky jsou větve. Tady bere algoritmus SPF data k počítání optimálních cest do každého cíle v síti. Nejlepší cesta (=nejmenší náklady) je zapsána do směrovací tabulky. Pokud existuje více cest se stejnými náklady do jednoho cíle, provoz se mezi ně rovnoměrně rozkládá.

4. EIGRP – “Hybridní“ směrovací protokol

EIGRP (Enhanced Interior Gateway Routing Protocol) je proprietární protokol firmy Cisco, což bohužel znamená, že množství informací o jeho funkcích není veřejně dostupných. Proto se většinou omezíme na popis chování protokolu, místo přesného popisu.

Protokol EIGRP vychází z předchozího úspěšného protokolu IGRP, který se řadí do skupiny distance-vector protokolů. EIGRP je mu v mnohém podobný (například ve výpočtu metriky), ale také se v mnohém značně liší. Značně využívá funkce typické pro skupinu link-state protokolů. Tato kombinace link-state a distance-vector je důvodem, proč se nazývá *hybridním* či *speciálním* směrovacím protokolem, zkrátka nejde zařadit do typických skupin. Pokud už znáte RIP a OSPF, budou vám některé části EIGRP připadat velmi povědomé.

EIGRP je plně zpětně kompatibilní s IGRP, a také podporuje redistribuci cest z ostatních protokolů, konkrétně: OSPF, RIP, IS-IS, EGP, BGP.

4.1. Algoritmus DUAL

DUAL (Diffusing Update Algorithm Finite-State Machine) je stroj pro výpočet a porovnávání různých cest v síti. Kvůli proprietární nátuře EIGRP o jeho fungování mnoho nevíme. Jeho hlavní funkcí je porovnání různých cest v síti z topologické tabulky (o té později) a výběr té nejlepší cesty do každého cíle. Tato cesta je pak zapsána do směrovací tabulky.

4.2. Moduly a RTP

Protokol EIGRP byl vyvíjen v době poměrně často se měnících standardů, a proto byl kladen velký důraz na jeho nezávislosti na směrovaných informacích a protokolech. Důsledkem toho je na vnitřní komunikaci mezi směrovači používán protokol RTP (Reliable Transport Protocol), který podobně jako kombinace TCP/UDP umí jak spolehlivé, tak i nespolehlivé doručování zpráv.

Jak již bylo zmíněno, EIGRP se snaží být nezávislý na směrovaných protokolech. Proto obsahuje systém modulů, kde každý podporovaný protokol má svůj vlastní modul. Výhodou je, že tak lze

přidat další protokol bez nutnosti větších změn ve zbytku systému. Příkladem můžou být moduly pro směrování IPX, AppleTalk, ale nejvyužívanější je samozřejmě IP-EIGRP, kterému se budeme dále věnovat.

4.3. Data a jejich přenos v rámci EIGRP

EIGRP potřebuje (vzhledem ke své složitosti) k svému fungování poměrně velké množství dat. Sleduje mnoho informací o aktuálním stavu celé sítě, a proto je logické, že potřebuje robustní systém ukládání dat. Tyto informace jsou sdružené podle příbuznosti dat do několika typů tabulek. Funkce těchto typů tabulek jsou vysvětleny níže.

Za přenášení informací nacházejících se v těchto tabulkách mezi jednotlivými směrovači jsou zodpovědny specializované typy paketů. Vzhledem k rozdílné podstatě dat je použito několik typů paketů. Ty jsou podrobněji také popsány níže.

4.3.1. Tabulky

Jako zbytek protokolu EIGRP, i konkrétní funkce a struktury tabulek jsou proprietární, takže následující popis bude opět docela stručný. EIGRP využívá tří typů tabulek: tabulky sousedů, směrovací tabulky a tabulky síťové topologie.

4.3.1.1. Tabulka sousedů

Tabulka sousedů sleduje, s kterými ostatními směrovači v sítí je navázáno přímé sousedství (podobně jako u protokolu OSPF). Tato tabulka je naprosto kritická k fungování konvergence, a veškerých operací se směrováním. Každý směrovač má jednu tuto tabulku pro každý směrovaný protokol (ale my zde zabýváme jen směrováním IP, takže si pro zjednodušení můžete představit, že má pouze jednu tabulku celkem). Používá se taky při komunikaci na ověření, že pakety od sousedů chodí ve správném pořadí.

4.3.1.2. Směrovací tabulka

Směrovací tabulka obsahuje až 6 cest ke každému potenciálnímu cíli v síti, což je značně více než u ostatních algoritmů. Tyto cesty jsou vypočítány pomocí algoritmu DUAL. Každou změnu ve své směrovací tabulce směrovač okamžitě hlásí všem svým sousedům. Podobně jako u tabulky sousedů, každý směrovaný protokol má vlastní tabulku.

4.3.1.3. Tabulka síťové topologie

Tabulka síťové topologie je ze všech zde uvedených tabulek zdaleka ta nejrozsáhlejší. Obsahuje všechny informace o všech cestách v síti, které jsou potřeba pro vybrání optimální cesty do jakéhokoliv cíle v síti: šířku pásma, očekávané zpoždění, spolehlivost, úroveň zatížení, max. Velikost přenosové jednotky, vdálenost od sousedního směrovače, nejnižší vypočtenou vzdálenost a zdroj cesty (ID směrovače, používá se jen v případě, že cesta je převzata od jiného protokolu). Jako u předchozích tabulek, i zde má každý modul svoji tabulku. l

Na rozdíl od OSPF neskladuje celou topografii *sítě*, pouze všechny cesty, které dostal od ostatních směrovačů.

DUAL využívá data z tabulky síťové topologie pro výpočet cest v směrovací tabulce. Pokud je cesta tzv. Pasivní, potom vše funguje a daná cesta je stabilní. Do směrovací tabulky je propsán následník (nejbližší směrovač na cestě, podobné next-hop routeru v RIP) a přípustný následník (možná alternativa následníku.

Pokud je ovšem cesta v tzv. Aktivním stavu, znamená to, že ztratila oba následníky a probíhá nad ní přepočet. Přepočet je náročný na systémové prostředky a může trvat dlouho, proto se používají dva následníky a až po dvojitém selhání dojde k přepočtu. Při samotném procesu se směrovač ptá všech svých sousedů na informace o cestě potřebným směrem. Proces přepočtu končí, až dostane odpověď od všech sousedů, což vyústí buď generací nové cesty nebo označením cíle jako nedostupného.

4.3.2. Pakety

Známe 5 typů paketů na přenos interních informací v protokolu EIGRP, ale i zde narážíme na problém proprietárnosti, takže popis paketů bude velmi krátký.

4.3.2.1. Kontaktní Hello pakety

Zasílají si je sousedící směrovače, právě podle nich na počátku komunikace zjistí směrovač svoje sousedy. Následně se využívají pouze na ověření, že sousedé jsou aktivní a schopni odpovídat. Jsou posílány multicastem všude okolo.

4.3.2.2. Potvrzovací pakety

Potvrzovací pakety neobsahují žádné informace. Používají se pouze na potvrzení přijetí zprávy vysílajícímu směrovači, pokud to zpráva vyžaduje *(viz RTP)*.

4.3.2.3. Aktualizační pakety

Aktualizační pakety slouží pro předávání informací týkajících se topologie mezi směrovači. Můžou být použity pro zaslání celé topologie nově připojenému směrovači (podobně jako u OSPF), ale častěji jsou používány přo zasílání updatů všem sousedům při detekci změny topologie. Jsou zasílány vždy v spolehlivém módu *(viz RTP)*.

4.3.2.4. Paket dotazu a odpovědi

Funkce paketů dotazu a odpovědi je velmi intuitivní. Dotaz je zaslán sousedům s požadavkem na informace, odpověď je paket obsahující požadované informace. Tento systém je použit například když směrovač ztratí všechny cesty do určitého cíle *(viz Tabulka Síťové Topologie, aktivní stav cesty)*.

5. Závěr

Cílem práce bylo vytvoření materiálů pro podporu výuky dynamických směrovacích protokolů. Byla vypracována jak za pomoci znalostí předchozího studia s použitím Cisco Networking Academy tak i informací načerpaných z uvedené literatury. V tomto ohledu byl maturitní projekt vypracován relativně úspěšně. Jeho finální forma plní většinu bodů z původního zadání, největším nedostatkem je ovšem forma webové prezentace, která nebyla dokončena v daném časovém termínu.

Přesto autor doufá, že práce najde svoje využití v rozšiřování znalostí u zájemců o obor informatiky, ať už při výuce nebo jako zdroj pro samostudium dynamického směrování.

Citace:

MINAŘÍK, Michal. *Dynamické směrování* [online]. [cit. 2024-03-29]. Dostupné z: https://dsn.felk.cvut.cz/wiki/_media/vyuka/cviceni/y36sps/semestralky/dynamicke_smerovani.pdf. Seminární práce. ČVUT.

KMONÍČEK, Tomáš. Analýza principů IGP a EGP routovacích protokolů [online]. 2013 [cit. 2024-03-31]. Dostupné z: https://dk.upce.cz/bitstream/handle/10195/51622/KmonicekT_AnalyzaPrincipu_JH_2013.pdf?sequence=2&isAllowed=y. Bakalářská práce. Univerzita Pardubice.

TCP/IP - Routing - směrování. SAMURAJ-cz [online]. [cit. 2024-04-02]. Dostupné z: https://www.samuraj-cz.com/clanek/tcpip-routing-smerovani/

Autonomní systém. Wikipedie [online]. [cit. 2024-03-29]. Dostupné z: https://cs.wikipedia.org/wiki/Autonomn%C3%AD_syst%C3%A9m

Směrovací protokol. *Wikipedie* [online]. [cit. 2024-03-29]. Dostupné z: https://cs.wikipedia.org/wiki/Sm%C4%9Brovac%C3%AD_protokol

GRYGAREK, Petr. *Směrování a směrovací algoritmy* [online]. [cit. 2024-03-29]. Dostupné z: https://www.cs.vsb.cz/grygarek/PS/lect/propojovani_smerovace_smerovani.html

Routing Information Protocol. *Wikipedia* [online]. [cit. 2024-03-29]. Dostupné z: https://en.wikipedia.org/wiki/Routing_Information_Protocol

ChatGPT. *Použito jako zdroj informací (Classful routing)* [online]. [cit. 2023-11-21]. Dostupné z: https://chat.openai.com

Malkin, G., "RIP Version 2 - Carrying Additional Information", RFC 1723, DOI 10.17487/RFC1723, November 1994, <https://www.rfc-editor.org/info/rfc1723>.

Malkin, G., "RIPng Protocol Applicability Statement", RFC 2081, DOI 10.17487/RFC2081, January 1997, <https://www.rfc-editor.org/info/rfc2081>.

MOY, J. IETF. OSPF Version 2. Internet Engineering Task Force (IETF) [online]. [cit. 2024-03-29]. Dostupné z: https://www.ietf.org/rfc/rfc2328.txt

Dijkstra's algorithm. Online. Wikipedia. Dostupné z: https://en.wikipedia.org/wiki/Dijkstra%27s_algorithm. [cit. 2024-03-29].

JIROVSKÝ, Lukáš. Teorie grafů ve výuce na střední škole. Teorie grafů [online]. [cit. 2024-03-29]. Dostupné z: https://teorie-grafu.cz/uvod/vyuziti-grafu.php

SPORTACK, Mark A. *Směrování v sítích IP: [autorizovaný výukový průvodce : samostudium : kompletní zdroj informací o směrování a protokolech v sítích IP]*. Cisco systems. Brno: Computer Press, 2004. ISBN isbn80-251-0127-4.

Diffusing update algorithm. *Wikipedia* [online]. [cit. 2024-03-29]. Dostupné z: https://en.wikipedia.org/wiki/Diffusing_update_algorithm

GRYGAREK, Petr. *Směrovací protokol OSPF* [online]. [cit. 2024-03-29]. Dostupné z: https://www.cs.vsb.cz/grygarek/SPS/lect/OSPF/ospf.html

Packet types for OSPF. IBM Documentation [online]. [cit. 2024-03-29]. Dostupné z: https://www.ibm.com/docs/en/i/7.1?topic=concepts-packet-types-ospf

IETF. Cisco's Enhanced Interior Gateway Routing Protocol (EIGRP). Internet Engineering Task Force (IETF) [online]. 2016 [cit. 2024-03-29]. Dostupné z: https://datatracker.ietf.org/doc/html/rfc7868

EIGRP Configuration. GeeksforGeeks [online]. [cit. 2024-03-29]. Dostupné z: https://www.geeksforgeeks.org/eigrp-configuration/

EIGRP Configuration Step-by-Step Guide. ComputerNetworkingNotes [online]. [cit. 2024-03-29]. Dostupné z: https://www.computernetworkingnotes.com/ccna-study-guide/eigrp-configuration-step-by-step-guide.html

Configuring OSPF. CISCO [online]. [cit. 2024-03-29]. Dostupné z: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/iproute_ospf/configuration/xe-16/iro-xe-16-book/iro-cfg.html

ChatGPT. *Použito jako zdroj informací / formátování konfiguračních příkazů v přílohách* [online]. [cit. 2024-03-30]. Dostupné z: https://chat.openai.com

OSPFv3 Configuration Example. *CISCO* [online]. [cit. 2024-03-29]. Dostupné z: https://www.cisco.com/c/en/us/support/docs/ip/ip-version-6-ipv6/112100-ospfv3-config-guide.html#toc-hId--1070221606

RIP + IPv4

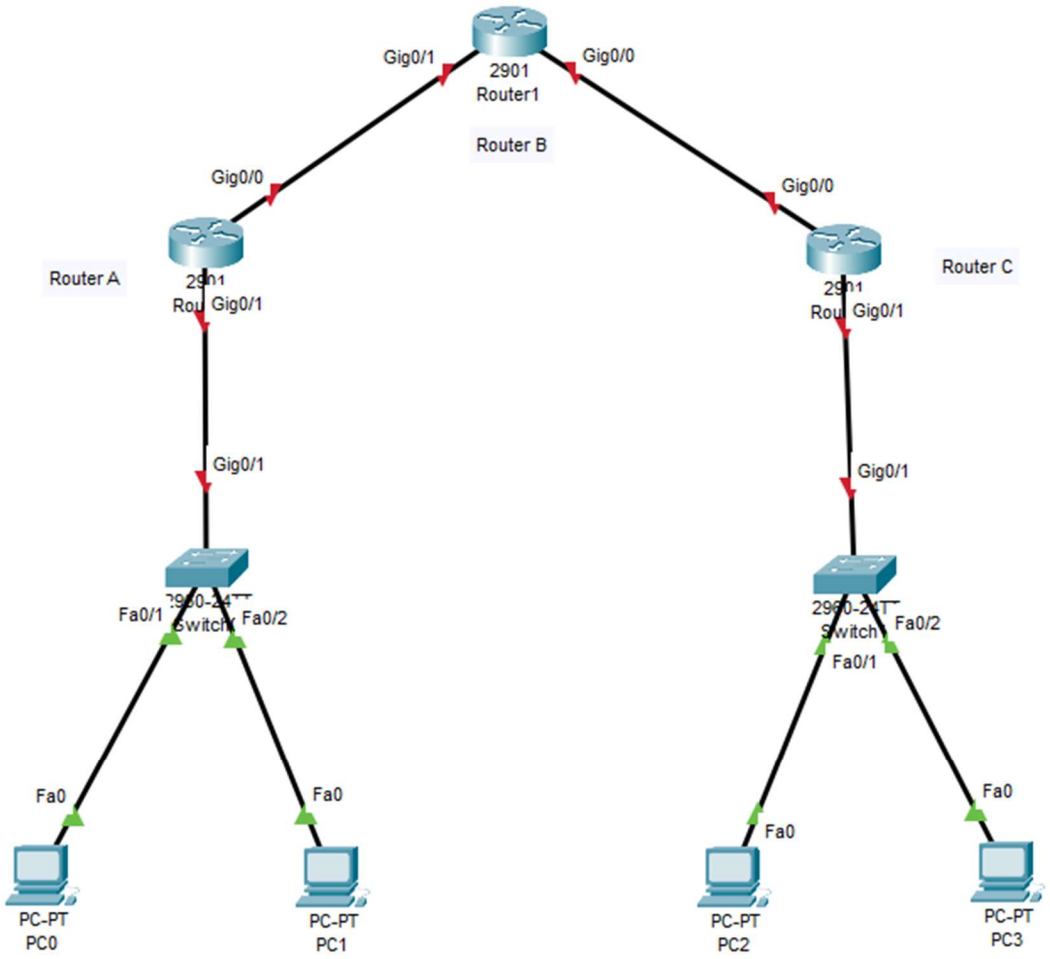
Zařízení:

3x 2901 router

2x 2960 switch

4x PC

Topologie:



Sítě:

- LAN 1 (pc 0 a 1)

    - 192.168.10.0/24
    - PC 0: 192.168.10.11
    - PC 1: 192.168.10.12
    - Default gateway (router A/R1): 192.168.10.1
- LAN 2 (pc 2 a 3)
    - 192.168.20.0/24
    - PC 2: 192.168.20.21
    - PC 3: 192.168.20.22
    - Default gateway (router C/R3): 192.168.20.1
- Point-to-point A-B
    - 10.0.0.0/28
    - Router R1: 10.0.0.2
    - Router R0: 10.0.0.1
- Point-to-point B-C
    - 10.0.20.0/28
    - Router R0: 10.0.20.1
    - Router R2: 10.0.20.2

RIP konfigurace

Router A:

```
RouterA> enable
RouterA# configure terminal
RouterA(config)# router rip
RouterA(config-router)# network 192.168.10.0
RouterA(config-router)# network 10.0.0.0
RouterA(config-router)# exit
RouterA(config)# exit
```

Router B:

```
RouterB> enable
RouterB# configure terminal
RouterB(config)# router rip
RouterB(config-router)# network 10.0.0.0
RouterB(config-router)# network 10.0.20.0
RouterB(config-router)# exit
RouterB(config)# exit
```

Router C:

```
RouterC> enable
RouterC# configure terminal
RouterC(config)# router rip
RouterC(config-router)# network 192.168.20.0
RouterC(config-router)# network 10.0.20.0
RouterC(config-router)# exit
RouterC(config)# exit
```

RIP + IPv6

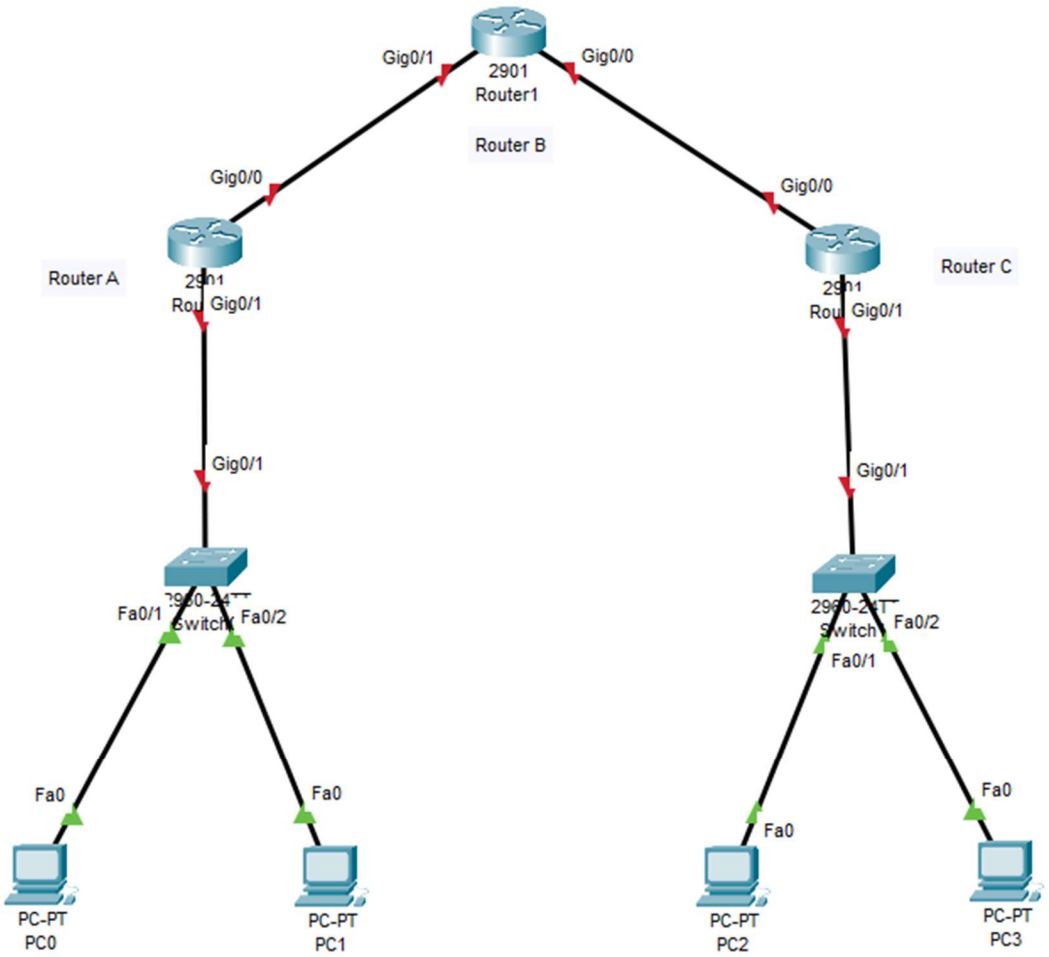
Zařízení:

3x 2901 router

2x 2960 switch

4x PC

Topologie:



Sítě:

- LAN 1 (pc 0 a 1)

    - 2001:db8:0:1::/64
    - PC 0: 2001:db8:0:1::11
    - PC 1: 2001:db8:0:1::12
    - Default gateway (router A): 2001:db8:0:1::1
- LAN 2 (pc 2 a 3)
    - 2001:db8:0:2::/64
    - PC 2: 2001:db8:0:2::21
    - PC 3: 2001:db8:0:2::22
    - Default gateway (router C): 2001:db8:0:2::1
- Point-to-point A-B
    - 2001:db8:0:10::/127
    - Router A: 2001:db8:0:10::0
    - Router B: 2001:db8:0:10::1
- Point-to-point B-C
    - 2001:db8:0:11::/127
    - Router B: 2001:db8:0:11::0
    - Router C: 2001:db8:0:11::1

RIPng konfigurace

Použijeme název procesu “NET”.

Router A:

```
RouterA> enable
RouterA# configure terminal
RouterA(config)# ipv6 unicast-routing
RouterA(config)# ipv6 router rip NET
RouterA(config)# interface GigabitEthernet0/1
RouterA(config-if)# ipv6 rip NET enable
RouterA(config-if)# exit
RouterA(config)# interface GigabitEthernet0/0
RouterA(config-if)# ipv6 rip NET enable
RouterA(config-if)# exit
RouterA(config)# exit
```

Router B:

```
RouterB> enable
RouterB# configure terminal
RouterB(config)# ipv6 unicast-routing
RouterB(config)# ipv6 router rip NET
RouterB(config)# interface GigabitEthernet0/1
RouterB(config-if)# ipv6 rip NET enable
RouterB(config-if)# exit
RouterB(config)# interface GigabitEthernet0/0
RouterB(config-if)# ipv6 rip NET enable
RouterB(config-if)# exit
RouterB(config)# exit
```

RouterC:

```
RouterC> enable
RouterC# configure terminal
RouterC(config)# ipv6 unicast-routing
RouterC(config)# ipv6 router rip NET
RouterC(config)# interface GigabitEthernet0/1
RouterC(config-if)# ipv6 rip NET enable
RouterC(config-if)# exit
RouterC(config)# interface GigabitEthernet0/0
RouterC(config-if)# ipv6 rip NET enable
RouterC(config-if)# exit
RouterC(config)# exit
```

EIGRP + IPv4

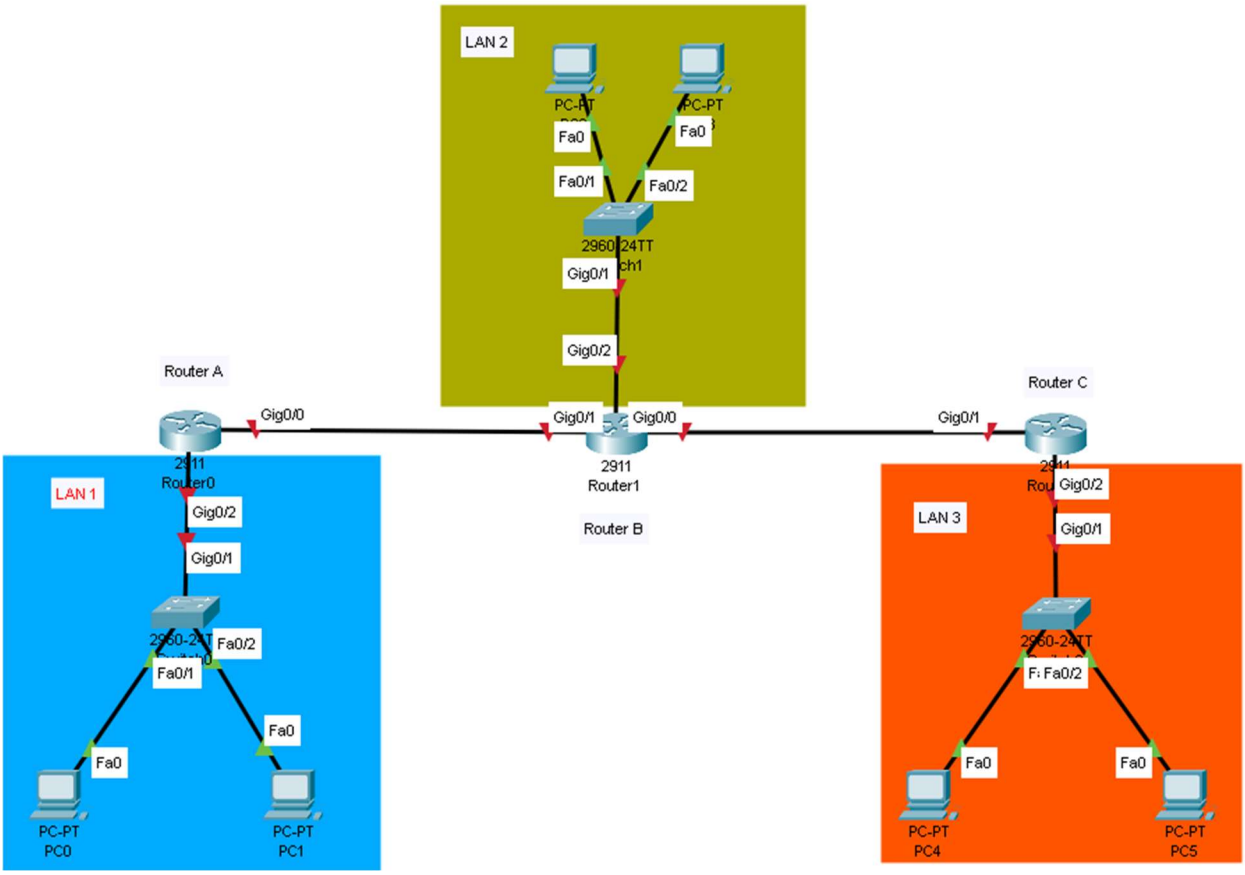
Zařízení:

3x router (2911)

3x switch (2960)

6x PC

Topologie:



Sítě:

- Lan 1: 192.168.1.0/24
  - Router A interface: 192.168.1.1
- Lan2: 192.168.2.0/24
  - Router B interface: 192.168.2.1
- Lan3: 192.168.3.0/24
  - Router C intrface: 192.168.3.1
- Point-to-point A-B: 10.0.0.0/30

    - Router A: 10.0.0.1
    - Router B: 10.0.0.2
- Point-to-point B-C: 10.0.0.4/30
    - Router B: 10.0.0.5
    - Router C: 10.0.0.6

EIGRP konfigurace

Použijeme číslo AS “100”. Toto se musí shodovat na všech routerech.

Router A:

```
RouterA> enable
RouterA# configure terminal
RouterA(config)# router eigrp 100
RouterA(config-router)# network 192.168.1.0
RouterA(config-router)# network 10.0.0.0
RouterA(config-router)# no auto-summary
RouterA(config-router)# exit
```

Router B:

```
RouterB> enable
RouterB# configure terminal
RouterB(config)# router eigrp 100
RouterB(config-router)# network 192.168.2.0
RouterB(config-router)# network 10.0.0.0
RouterB(config-router)# network 10.0.0.4
RouterB(config-router)# no auto-summary
RouterB(config-router)# exit
```

Router C:

```
RouterC> enable
RouterC# configure terminal
RouterC(config)# router eigrp 100
RouterC(config-router)# network 192.168.3.0
RouterC(config-router)# network 10.0.0.4
RouterC(config-router)# no auto-summary
RouterC(config-router)# exit
```

EIGRP + IPv6

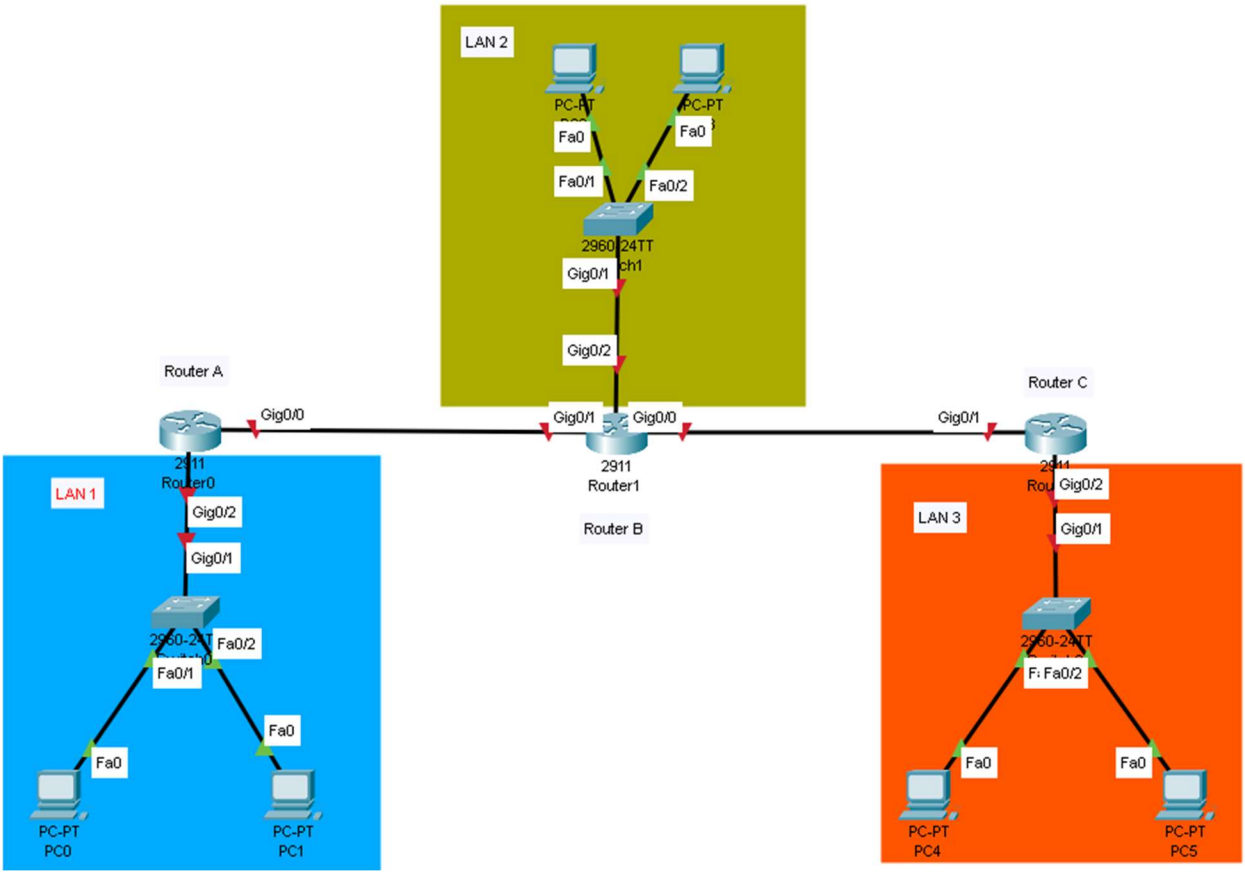
Zařízení:

3x router (2911)

3x switch (2960)

6x PC

Topologie:



Sítě:

- LAN 1: 2001:db8:0:1::/64
  - Router A interface: 2001:db8:0:1::1
- LAN 2: 2001:db8:0:2::/64
  - Router B interface: 2001:db8:0:2::1
- LAN 3: 2001:db8:0:3::1
  - Router C interface: 2001:db8:0:3::1
- Point-to-point A-B: 2001:db8:0:10::/127

    - Router A: 2001:db8:0:10::0
    - Router B: 2001:db8:0:10::1
- Point-to-point B-C: 2001:db8:0:11::/127
    - Router B: 2001:db8:0:11::0
    - Router C: 2001:db8:0:11::1

EIGRP konfigurace

ASN nastavíme na 100. Toto číslo musí být *stejné* u všech routerů.

Router ID na rozdíl od toho musí být *jedinečné* pro každý router. Udává se ve formátu X.X.X.X

Router A:

```
RouterA> enable
RouterA# configure terminal
RouterA(config)# ipv6 unicast-routing
RouterA(config)# ipv6 router eigrp 100
RouterA(config-rtr)# eigrp router-id 1.1.1.1
RouterA(config-rtr)# interface GigabitEthernet0/0
RouterA(config-if)# ipv6 eigrp 100
RouterA(config-if)# interface GigabitEthernet0/2
RouterA(config-if)# ipv6 eigrp 100
RouterA(config-if)# end
RouterA# copy running-config startup-config
```

Router B:

```
RouterB> enable
RouterB# configure terminal
RouterB(config)# ipv6 unicast-routing
RouterB(config)# ipv6 router eigrp 100
RouterB(config-rtr)# eigrp router-id 2.2.2.2
RouterB(config-rtr)# interface GigabitEthernet0/0
RouterB(config-if)# ipv6 eigrp 100
RouterB(config-if)# interface GigabitEthernet0/1
RouterB(config-if)# ipv6 eigrp 100
RouterB(config-if)# interface GigabitEthernet0/2
RouterB(config-if)# ipv6 eigrp 100
RouterB(config-if)# end
RouterB# copy running-config startup-config
```

Router C:

```
RouterC> enable
RouterC# configure terminal
RouterC(config)# ipv6 unicast-routing
RouterC(config)# ipv6 router eigrp 100
RouterC(config-rtr)# eigrp router-id 3.3.3.3
RouterC(config-rtr)# interface GigabitEthernet0/1
RouterC(config-if)# ipv6 eigrp 100
RouterC(config-if)# interface GigabitEthernet0/2
RouterC(config-if)# ipv6 eigrp 100
RouterC(config-if)# end
RouterC# copy running-config startup-config
```

OSPF + IPv4

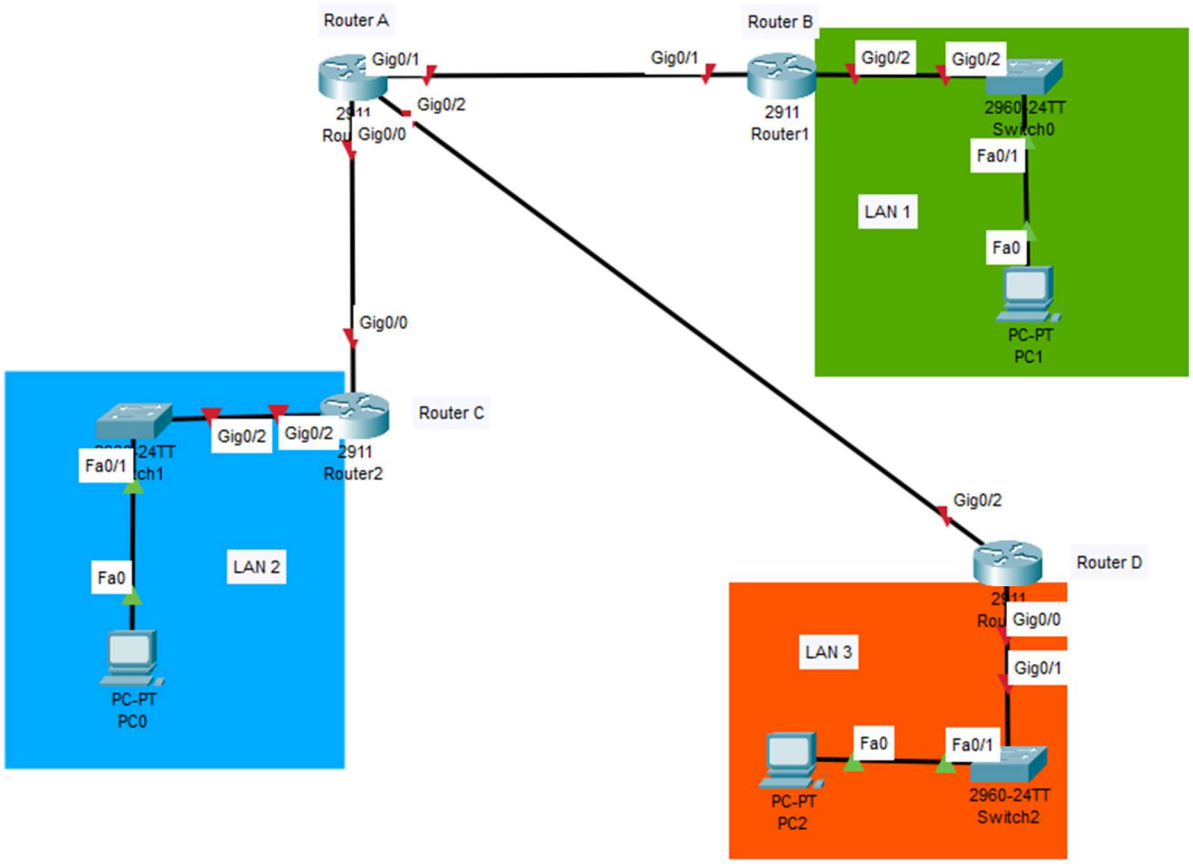
Zařízení:

4x router (2911)

3x switch (2960)

3x PC

Topologie:



Sítě:

- LAN 1: 192.168.1.0/24
  - Router B: 192.168.1.1/24
- LAN 2: 192.168.2.0/24
  - Router C: 192.168.2.1/24
- LAN 3: 192.168.3.0/24

    - Router D: 192.168.3.1/24
- P2P Network A-B: 10.1.1.0/30
    - Router A (Interface Gig0/1): 10.1.1.1/30
    - Router B (Interface Gig0/1): 10.1.1.2/30
- P2P Network A-C: 10.1.2.0/30
    - Router A (Interface Gig0/0): 10.1.2.1/30
    - Router C (Interface Gig0/0): 10.1.2.2/30
- P2P Network A-D: 10.1.3.0/30
    - Router A (Interface Gig0/2): 10.1.3.1/30
    - Router D (Interface Gig0/2): 10.1.3.2/30

OSPF konfigurace

Jako číslo procesu použijeme 1. Všechny routery dáme do oblasti (area) 0, protože naše síť je příliš malá na to, aby se vyplatilo ji dělit.

Pozor! Při zadávání adres sítí používáme wildcardové masky (opak subnetové masky).

Router A:

```
RouterA> enable
RouterA# configure terminal
RouterA(config)# router ospf 1
RouterA(config-router)# network 10.1.1.0 0.0.0.3 area 0
RouterA(config-router)# network 10.1.2.0 0.0.0.3 area 0
RouterA(config-router)# network 10.1.3.0 0.0.0.3 area 0
RouterA(config-router)# end
RouterA# copy running-config startup-config
```

Router B:

```
RouterB> enable
RouterB# configure terminal
RouterB(config)# router ospf 1
RouterB(config-router)# network 10.1.1.0 0.0.0.3 area 0
RouterB(config-router)# network 192.168.1.0 0.0.0.255 area 0
RouterB(config-router)# end
RouterB# copy running-config startup-config
```

Router C:

```
RouterC> enable
RouterC# configure terminal
RouterC(config)# router ospf 1
RouterC(config-router)# network 10.1.2.0 0.0.0.3 area 0
RouterC(config-router)# network 192.168.2.0 0.0.0.255 area 0
RouterC(config-router)# end
RouterC# copy running-config startup-config
```

Router D:

```
RouterD> enable
RouterD# configure terminal
RouterD(config)# router ospf 1
RouterD(config-router)# network 10.1.3.0 0.0.0.3 area 0
RouterD(config-router)# network 192.168.3.0 0.0.0.255 area 0
RouterD(config-router)# end
RouterD# copy running-config startup-config
```

OSPF + IPv6

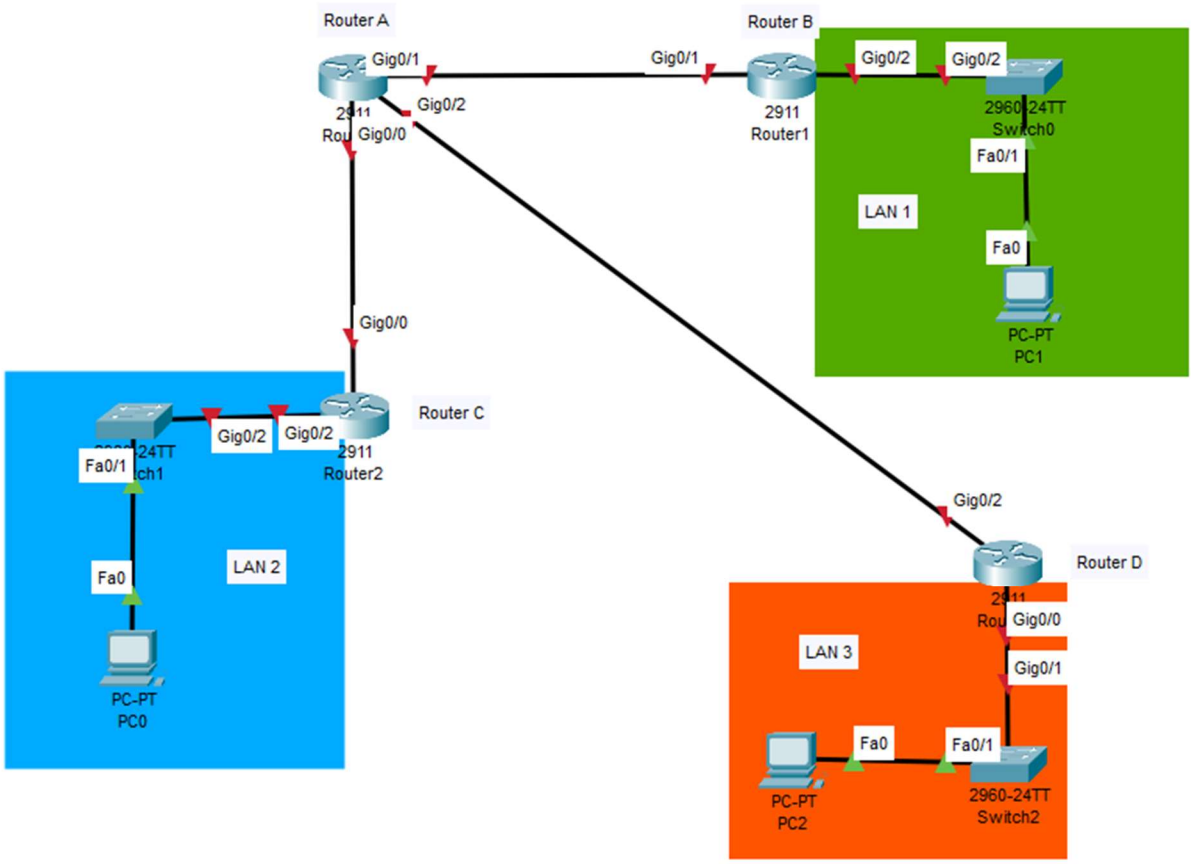
Zařízení:

4x router (2911)

3x switch (2960)

3x PC

Topologie:



Sítě:

- LAN 1: 2001:db8:1::/64
  - Router B: 2001:db8:1::1/64
- LAN 2: 2001:db8:2::/64
  - Router C: 2001:db8:2::1/64
- LAN 3: 2001:db8:3::/64

    - Router D: 2001:db8:3::1/64
- P2P Network A-B: 2001:db8:10:1::/127
    - Router A (Interface Gig0/1): 2001:db8:10:1::1/127
    - Router B (Interface Gig0/1): 2001:db8:10:1::2/127
- P2P Network A-C: 2001:db8:10:2::/127
    - Router A (Interface Gig0/0): 2001:db8:10:2::1/127
    - Router C (Interface Gig0/0): 2001:db8:10:2::2/127
- P2P Network A-D: 2001:db8:10:3::/127
    - Router A (Interface Gig0/2): 2001:db8:10:3::1/127
    - Router D (Interface Gig0/2): 2001:db8:10:3::2/127

OSPF konfigurace

Jako číslo procesu použijeme 1. Všechny routery dáme do oblasti (area) 0, protože naše síť je příliš malá na to, aby se vyplatilo ji dělit. Router ID musí být pokaždé unikátní.

Router A:

```
RouterA> enable
RouterA# configure terminal
RouterA(config)# ipv6 unicast-routing
RouterA(config)# ipv6 router ospf 1
RouterA(config-rtr)# router-id 1.1.1.1
RouterA(config-rtr)# exit
RouterA(config)# interface GigabitEthernet0/0
RouterA(config-if)# ipv6 ospf 1 area 0
RouterA(config-if)# exit
RouterA(config)# interface GigabitEthernet0/1
RouterA(config-if)# ipv6 ospf 1 area 0
RouterA(config-if)# exit
RouterA(config)# interface GigabitEthernet0/2
RouterA(config-if)# ipv6 ospf 1 area 0
RouterA(config-if)# exit
RouterA# copy running-config startup-config
```

Router B:

```
RouterB> enable
RouterB# configure terminal
RouterB(config)# ipv6 unicast-routing
RouterB(config)# ipv6 router ospf 1
RouterB(config-rtr)# router-id 2.2.2.2
RouterB(config-rtr)# exit
RouterB(config)# interface GigabitEthernet0/2
RouterB(config-if)# ipv6 ospf 1 area 0
RouterB(config-if)# exit
RouterB(config)# interface GigabitEthernet0/1
RouterB(config-if)# ipv6 ospf 1 area 0
RouterB(config-if)# exit
RouterB# copy running-config startup-config
```

Router C:

```
RouterC> enable
RouterC# configure terminal
RouterC(config)# ipv6 unicast-routing
RouterC(config)# ipv6 router ospf 1
RouterC(config-rtr)# router-id 3.3.3.3
RouterC(config-rtr)# exit
RouterC(config)# interface GigabitEthernet0/0
RouterC(config-if)# ipv6 ospf 1 area 0
RouterC(config-if)# exit
RouterC(config)# interface GigabitEthernet0/2
RouterC(config-if)# ipv6 ospf 1 area 0
RouterC(config-if)# exit
RouterC# copy running-config startup-config
```

Router D:

```
RouterD> enable
RouterD# configure terminal
RouterD(config)# ipv6 unicast-routing
RouterD(config)# ipv6 router ospf 1
RouterD(config-rtr)# router-id 4.4.4.4
RouterD(config-rtr)# exit
RouterD(config)# interface GigabitEthernet0/0
RouterD(config-if)# ipv6 ospf 1 area 0
RouterD(config-if)# exit
RouterD(config)# interface GigabitEthernet0/2
RouterD(config-if)# ipv6 ospf 1 area 0
RouterD(config-if)# exit
RouterD# copy running-config startup-config
```