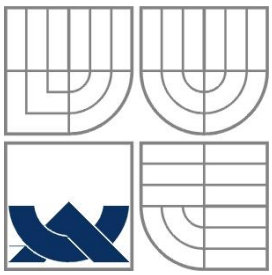
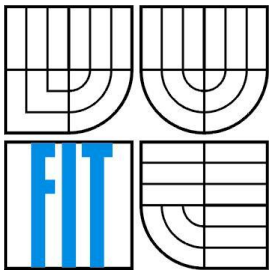


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

ARCHITEKTURY SYSTÉMŮ NA INTERNETU SE SKUPINOVÝM ADRESOVÁNÍM

ARCHITECTURES OF INTERNET-BASED SYSTEMS WITH MULTICASTING

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — Bc. Vladimír VESELÝ
AUTHOR

VEDOUCÍ PRÁCE — prof. Ing. Miroslav ŠVÉDA, CSc.
SUPERVISOR

BRNO 2009

Zadání

Architektury systémů na Internetu se skupinovým adresováním

Architectures of Internet-Based Systems with Multicasting

Vedoucí:

Švéda Miroslav, prof. Ing., CSc., UIFS FIT VUT

Přihlášen:

Veselý Vladimír, Bc.

Zadání:

Seznamte se s komunikačními protokoly souvisejícími se skupinovým adresováním na úrovni IP na Internetu.

1) Navrhněte koncepci prostředí pro experimentování se skupinovým adresováním na intranetu.
2) Navrhněte architekturu prostředí a základních nástrojů pro experimentování se skupinovým adresováním na intranetu/Internetu.
3) Vytvořte prototyp takového prostředí.
4) Navrhněte vhodné testovací úlohy a ověřte funkčnost jak jednotlivých nástrojů, tak i jejich spolupráce v rámci prostředí.
5) Zhodnoťte řešení a navrhněte případné pokračování projektu.

Část požadovaná pro obhajobu SP:

Splnění prvních 3 bodů zadání.

Kategorie:

Počítačové sítě

Literatura:

- Deering, S.E.: Host Extensions for IP Multicasting, IETF RFC 1112, New York (August 1989).
- Fenner, W.: Internet Group Management Protocol, Version 2, IETF RFC 2236, New York (November 1997).
- Puzmanova, R.: Routing and Switching, Time of Convergence? Addison-Wesley, 2003.
- Miller C.K.: Multicast Networking and Applications, Addison-Wesley, 1999.

Abstrakt

Spolu s rozvojem šíření multimediálního obsahu a aplikací s distribuovaným výpočtem napříč Internetem vzrostla důležitost optimalizace provozu, zde se jako nejlepší nástroj k dosažení cíle ukázal multicasting. Tato diplomová práce na téma skupinového směrování v Internetu klade důraz zejména na protokoly multicastem užívané, přičemž se snaží shrnuté poznatky aplikovat v konkrétním komerčním řešení a ozkoušet je tak v praxi s výstupem v podobě podrobného rozboru činnosti.

Klíčová slova

Multicast, multicasting, skupinové směrování, IGMP, IGMP snooping, PIM, PIM-SM, PIM-DM, PIM-SDM, CGMP, RGMP, MSDP, MOSPF, M-BGP, MBONE, Anect, Cisco, VLC Media Player, 224.0.0.13

Abstract

With rapid expansion of interest in multimedia and distributed computing applications across the Internet increases importance of optimized delivery of group traffic. According to current situation the best practice to achieve this goal is multicasting. This master’s thesis summarizes multicasting methods and facts. Also it draws conclusions from practical application of presented information on commercial project.

Keywords

Multicasting, multicast routing, IGMP, IGMP snooping, PIM, PIM-SM, PIM-DM, PIM-SDM, CGMP, RGMP, MSDP, MOSPF, M-BGP, MBONE, Anect, Cisco, VLC Media Player, 224.0.0.13

Architektury systémů na Internetu se skupinovým adresováním

Poděkování

Touto cestou bych rád poděkoval všem těm, bez kterých by světlo světa nikdy tato práce nespatřila!

Nejprve mým rodičům Jeleně a Vladimírovi za jejich lásku, výchovu a péči, díky které jsem se dostal až na práh inženýrského studia. Stejná slova patří i zbytku rodiny, bez jejichž vřelé podpory bych si své působení na vysoké škole nebyl schopen představit.

Dále můj vděk patří všem těm pedagogům kolem výzkumné skupiny NES@FIT, které jsem měl tu čest za poslední čtyři roky potkat a kteří mi umožnili objevit svět počítačových sítí, jenž mě do dnešních dní upřímně nepřestává fascinovat.

Poděkovat bych chtěl i Ing. Michalu Rapcovi, za jeho rady a svatou trpělivost, kterou se mnou měl, stejně jako za jeho lví podíl na sehnání prostředků, na kterých byla praktická část této práce realizována a ozkoušena. On a kolektiv brněnské pobočky firmy Anect a.s. mi umožnili nejen profesně dospět.

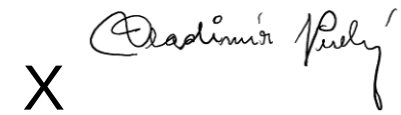
A v neposlední řadě bych tuto práci rád dedikoval největšímu avšak opomíjenému humanistovi šumperského kraje Mgr. Karlu Zárubovi.

Prohlášení

Tímto prohlašuji, že tuto diplomovou práci jsem vypracoval samostatně pod vedením prof. Ing. Miroslava Švédy, CSc. Další cenné informace mi předali Ing. Michal Rapco a Ing. Vladimír Veselý.

Uvedl jsem všechny elektronické či literární prameny a publikace, ze kterých jsem čerpal. Svým vlastnoručním podpisem stvrzuji, že práce ani jakákoli její část není plagiátem a že jsem se při její tvorbě nezpronevěřil svému imatrikulačnímu slibu.

21.5.2009



X

Vladimír Veselý

Obsah

1 Úvod

1.1 Problematika

Internet již od doby svého vzniku ovlivnil mnohé aspekty našich životů. Změnily se a vznikly nové způsoby, jakým lidé spolu komunikují, obchodují, vzdělávají se či tráví svůj volný čas. S tím, jak v posledních letech rapidně narostl počet uživatelů, kteří mají přístup k velmi levným, avšak přesto vysokorychlostním linkám, stoupl společně s tím i zájem o multimediální komunikaci – ať už se jedná o přenos rádia, živé vysílání, video/audio konferenční hovory.

Typicky se jde o druh komunikace, kdy existuje jeden zdroj vysílající stejný obsah desítkám či stovkám zajímajících se příjemců. Čím více uživatelů zvyklých na kvalitní multimediální služby je, tím více je potřeba optimalizovat tento provoz v rámci síťového prostředí – zejména druhé a třetí vrstvy modelu ISO/OSI [1].

Prostředek k dosažení tohoto cíle v IP sítích existuje již relativně dlouho, a to v podobě multicastingu. Tento elaborát navazuje na převážně teoretickou diplomovou práci o multicastingu od Ing. Igora Stehury [2]. Bude se zabývat spíše praktičtějšími aspekty okolo tématu, ale rozšíří čtenářovy poznatky i v oblastech, které se do výše zmíněné závěrečné práce již nevešly. Popíše například některé další protokoly související se skupinovým adresováním. Kromě prohloubení poznatků dané tématiky se bude zabývat i na jejich základě vytvořenou netriviální a na reálné použití zaměřenou modelovou architekturou multicastové struktury a jejím podrobným popisem a rozborem činnosti.

1.2 Stručný obsah kapitol

V druhé kapitole se vymezují a rekapitulují obecné informace týkající se multicastingu, a jak jsou mu podřízeny linková, síťová a aplikační vrstva ISO/OSI modelu.

Třetí kapitola shrnuje poznatky o protokolech a metodologiích, které všechny s multicastingem souvisí, přičemž některé z nich jsou pak využity při praktické realizaci: IGMP, IGMP snooping, CGMP, PIM, RGMP, MSDP, M-BGP a MOSPF.

Ve čtvrté kapitole je popsáno zadání a samotná praktická realizace sítě, na které pak byly provedeny testy multicastingu, jejichž výsledky shrnuje kapitola pátá.

Kapitola šestá se zabývá možnostmi zapojení se k již existujícím multicastovým strukturám v Internetu.

Poslední sedmá kapitola pak shrnuje poznatky, které jsou v této práci prezentovány.

2 Multicasting

Multicasting se v IP obecně používá tehdy, když jeden zdroj rozesílá stejná data vícero příjemcům. Zjednodušeně např. video-streamingový server posílá paket s multimediálním obsahem na jednu speciální adresu a koncové stanice se rozhodnou, zda se zapíší k odběru obsahu zasílanému právě na tuto adresu.

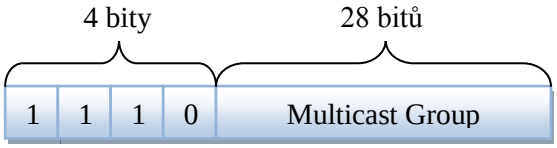
V současnosti je podporován jak v IP verze 4, tak i v IP verze 6, kde byla defakto komunikace pomocí broadcastu zrušena a byly naplno rozšířeny a využity možnosti směrování na základě příslušnosti k logické či konkrétní skupině.

2.1 Síťová vrstva

Síťová vrstva slouží k přenosu dat od zdroje k cíli skrz jednu či více sítí. Právě na této vrstvě jsou prováděny všechny rozhodnutí o směrování, kam budou data v následujícím kroku přeposlána. V rámci této vrstvy se hovoří o logickém adresování.

Multicastová komunikace se od unicastu (jeden zasílající vůči jednomu příjemci) či broadcastu (jeden zasílající vůči všem příjemcům v dané broadcastové doméně) liší na třetí vrstvě ISO/OSI právě v použití specializované IP adresy.

U IPv4 adresného prostoru je to vždy adresa z třídy D, přičemž rozložení bitů je ilustrováno:



Obrázek 1 (nastavení bitů adres třídy D)

Adresní rozsah je tedy v rámci tečkové decimální konvence zápisu v IPv4 od 224.0.0.0 až do 239.255.255.255, kde je ještě tento prostor rozdělen do tří logicky či dosahem odlišných skupin:

Viditelnost	Rozsah	Popis
Lokální	224.0.0.0 – 224.0.0.255	Rezervovány IANAou[1] pro použití různými lokálními síťovými protokoly (OSPF, NTP, RIP, EIGRP, aj.).
Globální	224.0.1.0 – 224.0.1.255 224.0.2.0 – 238.255.255.255	První skupina patří ještě také k rezervovaným, a to významným službám (NTP), zbytek je alokován dynamicky napříč Internetem s globální viditelností.
Administrativní	239.0.0.0 – 239.255.255.255	Rezervovány pro interní použití uvnitř jednoho autonomního systému, stejně jako privátní IP adresy nejsou globálně použitelné či viditelné.

Tabulka 1 (rozdělení multicastových adres podle viditelnosti)

[1] *Internet Assigned Numbers Authority* - http://www.iana.org/

Následující tabulka ukazuje dělení dosahu multicastového adresného prostoru v IPv6 a případně mapování na IPv4, tak jak jej definuje RFC 2365 [3] a navazující dokumenty:

Dosah	TTL	IPv6	IPv4
Rozhraní	0	FF01:	
Linka	1	FF02:	224.0.0.0 – 224.0.0.255
Lokalita	<32	FF05:	239.252.0.0 – 239.255.255.255
Organizace	≤255	FF08:	239.192.0.0 – 239.251.255.255
Globální	≤255	FF0E:	224.0.1.0 – 238.255.255.255
Rezervováno		FF0F:	239.0.0.0 – 239.191.255.255

Tabulka 2 (porovnání IPv6 a IPv4 multicastových adres podle dosahu)

Seznam všech rezervovaných adres a k nim přiřazených protokolů, stejně tak i seznamy adres globálních a k nim příslušejících aplikací, udržuje organizace IANA na adrese:

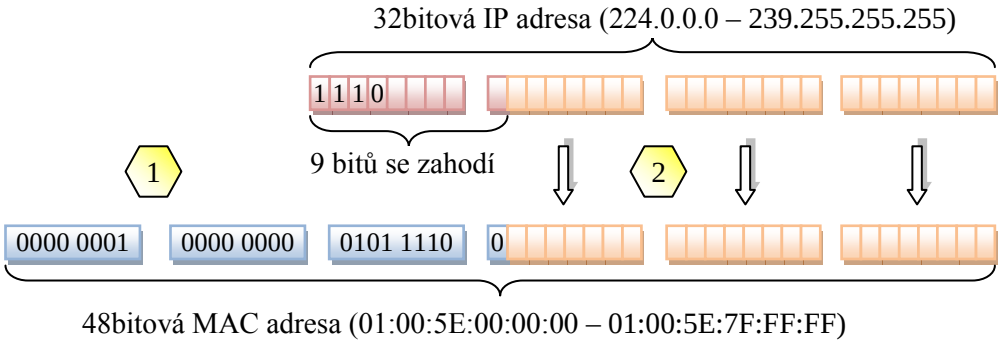
http://www.iana.org/assignments/multicast-addresses

2.2 Linková vrstva

Linková vrstva slouží k přenosu dat mezi dvěma sousedními síťovými body, v rámci této úrovně se hovoří o fyzickém adresování. Na této vrstvě funguje pro IPv4 protokol Address Resolution Protocol ARP [4], který pomáhá v překladu z logických adres na fyzické, kde je obvykle explicitně mapována jedna IPv4 adresa k jednomu síťovému rozhraní, které má jednu MAC adresu.

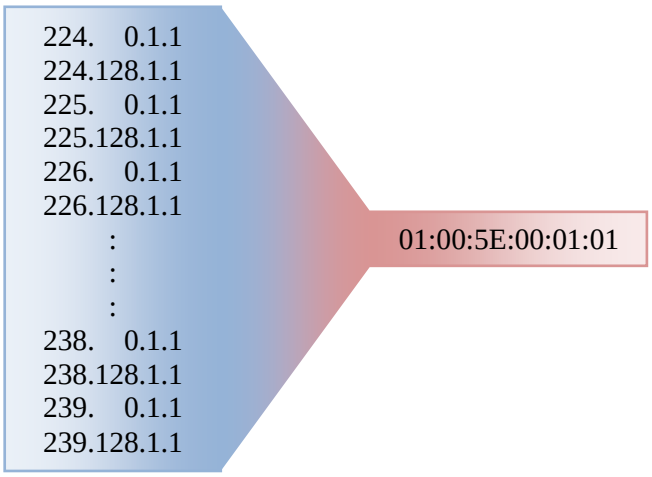
ARP však nelze použít v případě multicastingového provozu. Místo něho se používá automatizovaného algoritmu k převodu IP na MAC, tak jak popisuje následující text a Obrázek 2:

1) prvních 25 bitů namapované MAC adresy je vždy stejných (00000001 00000000 01011110 0);
2) zbývajících 23 bitů je tvořeno posledními 23 bity původní multicastové adresy.



Obrázek 2 (mapování multicastingové IP adresy na MAC adresu)

Díky tomu, že původní IPv4 adresa je 32bitová a použitá část z ní je jen 23bitová, tak dochází ke známému problému, kdy se na jednu namapovanou MAC adresu váže 32 adres z třídy D.



Obrázek 3 (korespondence mezi více IP a jednou MAC)

Pokud tak máme dva různé uživatele, kdy jeden je přihlášen k odběru z adresy 225.0.1.1 a druhý z adresy 225.128.1.1, tak oba budou dostávat toky dat určené pro obě tyto adresy a ještě 30 adres zbývajících – všechny namapované na stejnou MAC adresu.

V případě IPv6 je alternativou k ARP protokol Neigbor Discovery Protocol NDP [5], ten se však při multicastingu i tak nepoužívá, neb podpora skupinového směrování je zahrnuto již implicitně v IPv6.

2.3 Aplikační vrstva

Podle typu směru komunikace můžeme multicastingové aplikace běžně rozdělit do tří skupin:

- **One-to-Many** = jeden vysílač zasílá data více příjemcům (typicky využíváno pro přenos hlasu či obrazu);
- **Many-to-Many** = příjemci dat mohou být zároveň i vysílače (typicky aplikace pro distribuované výpočty či zpracování dat);
- **Many-to-One** = když vícero vysílačů zasílá data jednomu příjemci (typicky aplikace ve finančnictví, pojišťovnictví či hromadnému sběru dat).

3 Protokoly a metodologie

3.1 Internet Group Management Protocol

Jedná se o protokol operující na vršku síťové vrstvy, funkčností srovnatelný s ICMP pro unicasty v sítích IPv4. Prodělal od svého vzniku velký kus cesty, aktuální je verze IGMPv3, kterou pevně definuje dokument RFC 3376 [6] s návazností na nejlepší praktiky uváděné v RFC 4604 [7]. Pro pochopení dalších zmiňovaných protokolů pro správu skupin je znalost tohoto klíčová, proto začneme u popisu protokolů právě s ním.

IGMP je komunikačním protokolem mezi stanicí a směrovačem/přepínačem, který umožňuje:

- stanicím se selektivně přihlašovat a odhlašovat od odběru dat z dané multicastingové skupiny;
- směrovačům/přepínačům vyzvídat, jestli má někdo na daném síťovém segmentu zájem o odběr dat multicastingem obecně či z konkrétní skupiny.

Následující tabulka shrnuje typy zpráv a jejich význam, jenž IGMPv3 podporuje:

Typ	Kód	Zasílá	Popis
Membership Query	0x11	Router Switch	Informační zpráva, jestli má někdo na daném síťovém segmentu zájem odebírat nějaký multicast.
Version 3 Membership Report	0x22	Host	Přihlášení se k odběru od konkrétního zdroje v rámci specifické skupiny
Version 1 Membership Report	0x12	Host	Přihlášení se k odběru od specifické skupiny
Version 2 Membership Report	0x16	Host	Přihlášení se k odběru od specifické skupiny v2
Version 2 Leave Group	0x17	Host	Odhlášení se od specifické skupiny

Tabulka 3 (druhy IGMP zpráv)

Navíc zpráva *IGMP Membership Query* existuje ve třech funkčních variantách podle verze a i když má stejný Kód, tak ji jsou schopny způsobem příslušejícím verzi klienta zpracovávat všechny IGMP stanice:

- *General Query* (IGMPv1) – obecný dotaz na odběr jakýchkoli multicastových dat;
- *Group-Specific Query* (IGMPv2) – dotaz na odběr dat v paketu dále specifikované multicastingové skupiny;
- *Group-and-Source-Specific Query* (IGMPv3) – dotaz na odběr dat v paketu specifikované multicastingové skupiny a zdrojů, které ji poskytují.

Od verze IGMPv2 je i hierarchie v zasílání *IGMP Membership Query* zpráv, pokud existuje na síti více směrovačů. Vysílá vždy jeden, a to ten, který má nejmenší IP adresu, ostatní routry mlčí. Způsob, jakým se mezi sebou domlouvají, je čistě pasivní; pokud směrovač zachytí *IGMP Membership Query* zprávu od směrovače s nižší IP adresou, automaticky pozastaví své posílání zpráv tohoto typu, a to do vypršení časovače 400 sekund (přičemž každá další zpráva *IGMP Membership Query* s nižší IP adresou resetuje tento čítač).

Alternativou k IGMP v sítích IPv6 je protokol Multicast Listener Discovery s nejaktuálnější verzí MLDv2, který je definován v RFC 3810 [8] a jenž je podčástí protokolu ICMPv6, do kterého je plně integrován.

3.2 IGMP snooping

IGMP snooping (vznikl z nejlepší známé metodiky uváděné v RFC 4541 [9]) je technologie, která řeší obvyklé chování přepínačů, pokud dostanou nějaký multicastový rámec. Implicitně ho totiž zpracovávají jako broadcast, to znamená, že ho kopírují a odešlou všemi porty kromě příchozího. Ovšem v případě multicastingu (s přihlédnutím k typu dat, na které se obvykle používá) toto jednoduché avšak logické chování může v rozsáhlejších sítích činit značné problémy se zahlcováním prostředků.

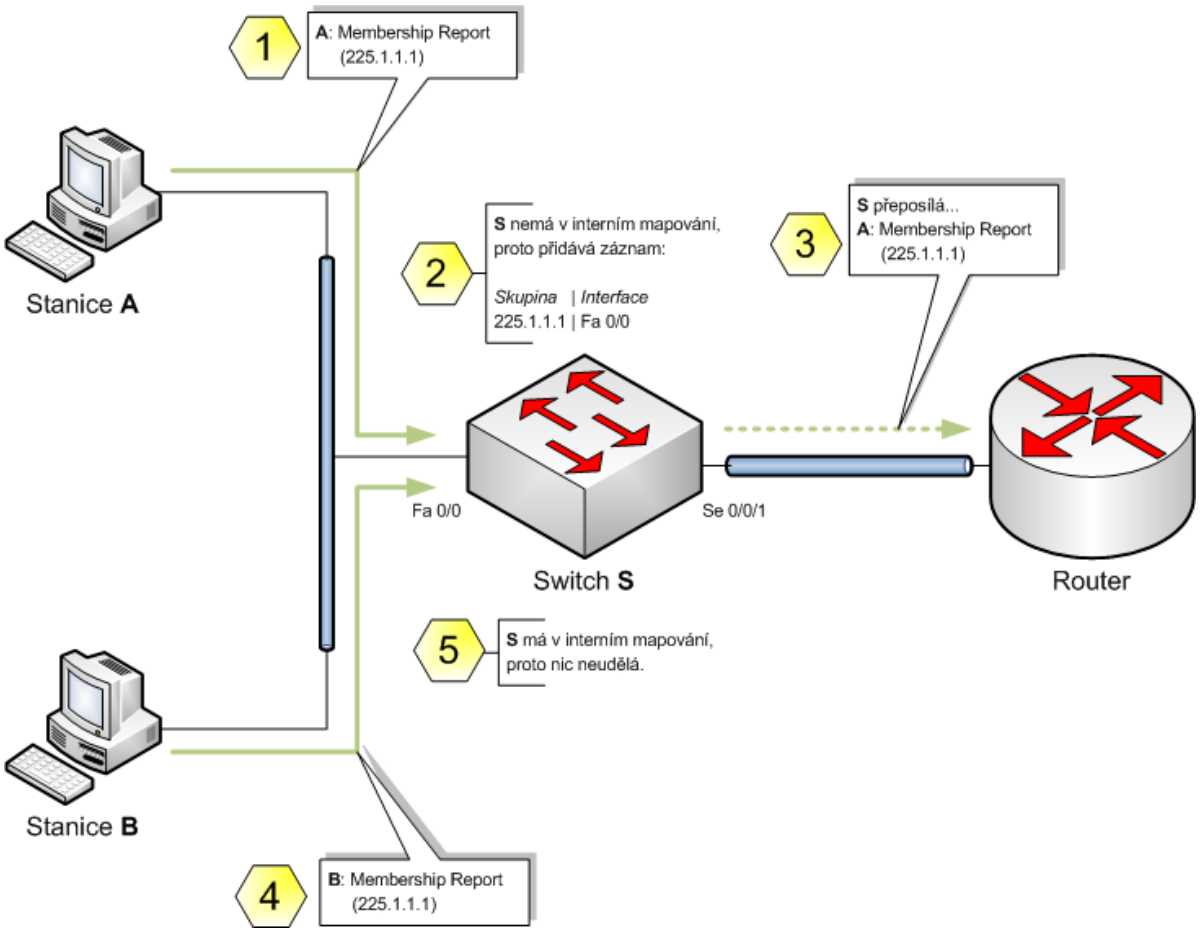
Jak už název napovídá, tak IGMP snooping (= anglicky slídit) je krom obvyklého zkoumání L2 rámců schopen se dívat i do hlaviček L3 paketů a zaznamenávat si případný IGMP provoz a pak určovat příslušnosti jednotlivých svých portů buď k lince, na které je směrovač, nebo koncové stanice.

Například přepínač od firmy Cisco [10] detekuje porty, které sousedí s multicasting zpracovávajícím směrovačem tak, že kontroluje, jestli z linky připojené na daný port nepřijde nějaká z následujících zpráv:

- *IGMP Membership Query* (zasílaná na 01:00:5E:00:00:01);
- *PIMv1 Hello* (zasílaná na 01:00:5E:00:00:02);
- *PIMv2 Hello* (zasílaná na 01:00:5E:00:00:0D);
- DVMRP probes (zasílané na 01:00:5E:00:04);
- *MOSPF Group-Membership-LSA* (zasílaný na 01:00:5E:00:05 nebo 01:00:5E:00:06).

Jak metodologie popisuje, tak v rámci připojování (resp. odpojování) nějaké stanice existují dvě alternativy chování, a to podle toho, je-li stanice první (resp. poslední) v pořadí k odběru dat z dané multicastingové skupiny. Scénář připojování ilustruje Obrázek 4 a k němu příslušející text:

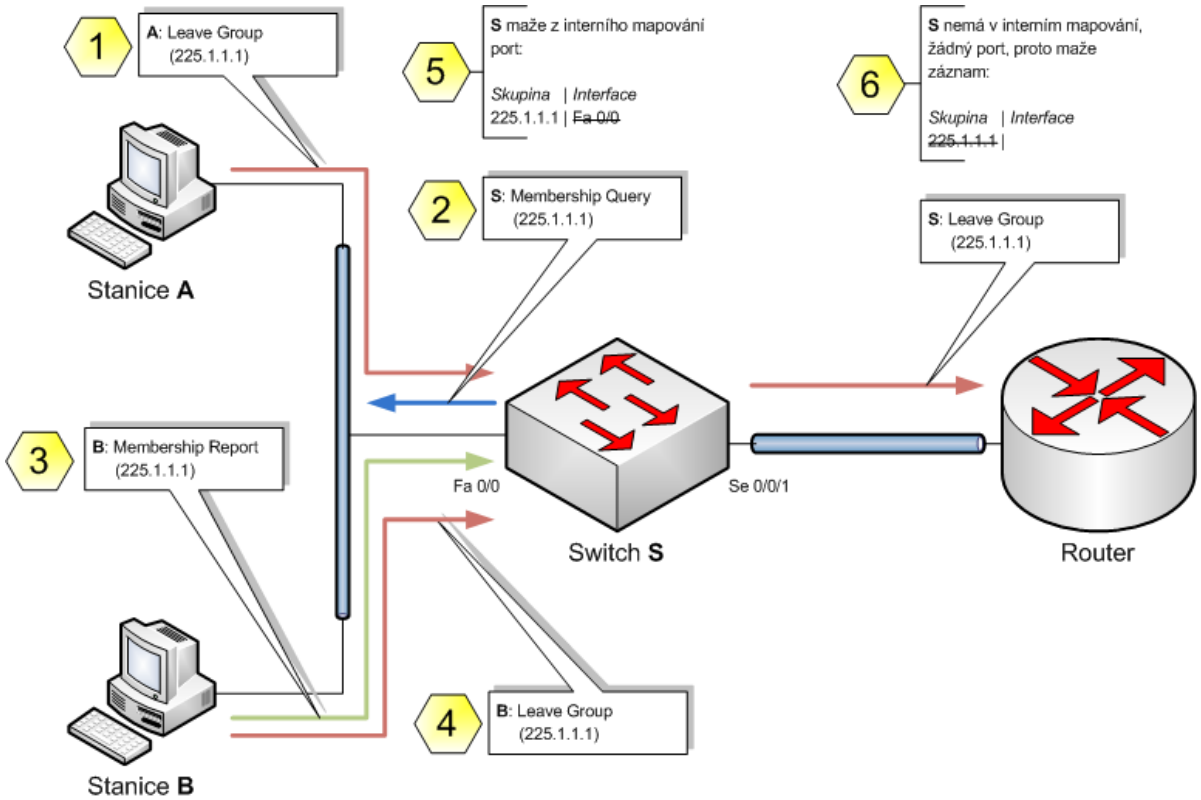
1) Stanice **A** je první, kdo se na daném síťovém segmentu připojuje k odběru multicastingových dat z nějaké skupiny, proto vysílá nevyžádaný *IGMP Membership Report*;
2) Switch **S** zachytil zprávu typu *IGMP Membership Report*, a protože je první na lince, vytváří si pro něj interní záznam, ve kterém je mapován daný port k požadované multicastový skupině, takže bude veškerá data určena pro tuto skupinu zasílat nově i na tento port;
3) **S** dále přepošle *IGMP Membership Report* od **A** na všechny své porty s routry;
4) Stanice **B** se nově taktéž rozhodne odebírat data z nějaké multicastové skupiny, a vyšle proto svůj *IGMP Membership Report*;
5) **S** jen porovná od **B** příchozí *IGMP Membership Report* a jen pokud ve svém interním záznamu nemá požadované multicastingové skupině přiřazen daný port, zopakuje kroky 2) a 3) pro výzvu **B**.



Obrázek 4 (IGMP snooping při připojování k multicastingu)

Scénář odpojování klientů od multicastingu pak ilustruje Obrázek 5 a následující text:

1) **A** chce opustit skupinu (ovšem **B** je stále registrována k odběru dat na stejném síťovém segmentu), a tak zasílá zprávu *IGMP Leave Group*;
2) **S** zachytí zprávu a ihned na to na linku, ze které se o záměru odhlašující se stanice dozvědělo, vyšle vlastní zprávu *IGMP Membership Query*, kterou zjišťuje, pokud zbyl ještě nějaký odběratel pro danou multicastingovou skupinu;
3) **B** odpovídá na výzvu **S** zprávou *IGMP Membership Report*, tím pádem nedojde k výmazu rozhraní z interního mapování multicastingových skupin k rozhraním;
4) **B** (poslední, kdo se na daném segmentu sítě chce odhlásit od odběru dat z dané skupiny) vyšle zprávu *IGMP Leave Group*;
5) **S** opakuje krok 2), ovšem nikdo na jeho výzvu *IGMP Membership Query* už neodpovídá, takže vymaže interně uchované mapování portu k dané odhlašované skupině;
6) V případě, že pro danou skupinu už neexistuje žádné interní mapování na porty (multicastová data k odběru totiž mohou být odebírána jiným segmentem sítě, který je k **S** připojen), ruší **S** záznam o celé skupině a zasílá všem svým routerům zprávu *IGMP Leave Group*.



Obrázek 5 (IGMP snooping při odpojování od multicastu)

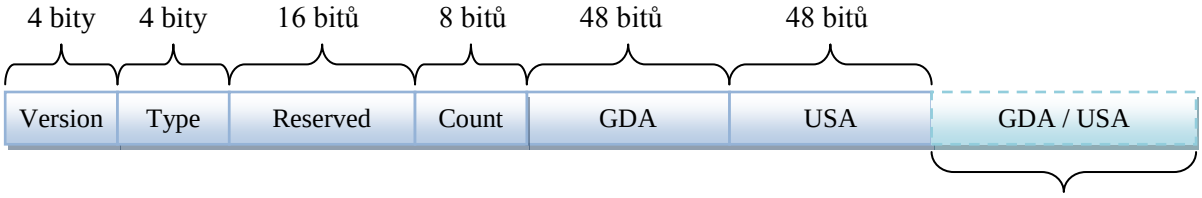
Proces zkoumání paketů při IGMP snoopingu, je však náročný na výpočetní prostředky přepínače, který je primárně koncipován jako zařízení určené pro L2 vrstvu, a zbytečně se tak ubírá na jeho výkonu. I to bylo jedním z důvodů, proč výrobci přepínačů investovali do vlastního vývoje a přišli z řadou vlastních proprietálních protokolů, které tento nešvar částečně či úplně řešil v době, kdy HW vybavení switchů nemělo ještě takové možnosti.

3.3 Cisco Group Management Protocol

A jedním z těchto protokolů je CGMP od firmy Cisco Systems[2]. Idea, která za ním stojí je celkem prostá: „Proč by měl switch složitě prozkoumávat obsahy paketů ohledně IGMP provozu, když tyto informace o skupinovém směrování má v kompletní podobě router?“

CGMP je protokol postaven na modelu jednosměrné komunikace serveru (router) vůči klientovi (switch). Umožňuje tak směrovačům předávat přepínačem své informace o IGMP provozu klientů přihlášených k odběru pomocí speciálních paketů a ovlivňovat tak obsah jejich CAM (*content adressable memory*) tabulek.

Struktura CGMP paketu:



případné další 96bitové dvojice
Group Destination Address / Unicast Source Address

- **Version** – aktuálně existuje jen jedna verze, je tedy nastaveno na 0x1;
- **Type** – určuje druh CGMP zprávy (buď *Join* 0x0, nebo *Leave* 0x1);
- **Reserved** – vyhrazeno pro budoucí použití, implicitně nastaveno na 0x0;
- **Count** – určuje kolik dvojic GA / SA se v paketu nachází, minimálně vždy jedna;
- **Group Destination Address** – určuje multicastovou adresu skupiny;
- **Unicast Source Address** – určuje adresu zdroje;

CGMP pakety jsou zabaleny do CGMP rámců. Tyto rámce přepínač rozliší od ostatního provozu tak, že mají cílovou MAC adresu nastavenou na hodnotu 01:00:0C:DD:DD:DD, která je rezervovaná čistě pro použití CGMP.

[2] *Cisco Systems, Inc.* - http://www.cisco.com/

Kombinace proměnných Type, GDA a USA určují, jaký druh operace switch vykoná:

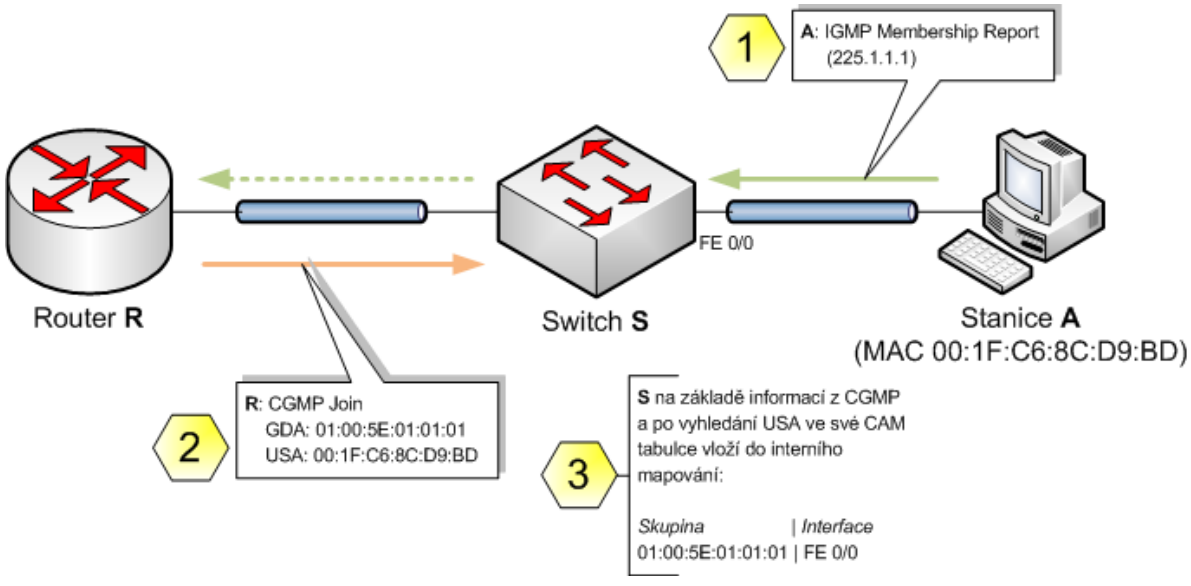
#	Typ	GDA	USA	Akce provedena přepínačem
1	Join	Multicastová MAC	MAC stanice	Přiřadí port dané multicastingové skupině
2	Leave	Multicastová MAC	MAC stanice	Smaže port dané multicastingové skupině
3	Join	00:00:00:00:00	MAC routeru	Switch zjistí, že na daném portu je router
4	Leave	00:00:00:00:00	MAC routeru	Switch zapomene, že na daném portu je router
5	Leave	Multicastová MAC	00:00:00:00:00	Vymaže z mapování specifikovanou skupinu
6	Leave	00:00:00:00:00	00:00:00:00:00	Vymaže všechna mapování pro všechny skupiny

Tabulka 4 (rozdělení CGMP zpráv podle funkce v návaznosti na vstupní parametry)

Uvažujme nyní situaci, kdy byl na směrovači povolen řídící protokol CGMP. V tomto případě začíná router na všechna svá rozhraní, na kterých je CGMP zapnuté, zasílat každých 60 vteřin zprávu **#3** (kterou uvádí Tabulka 4), čímž připojené CGMP přepínače zjišťují, že se na daném portu nachází router a že je stále aktivní.

Další v popisu fungování CGMP je zpracování žádosti klienta o připojení se do multicastingové skupiny, kterou popisuje následující text a Obrázek 6, přičemž předpokládáme, že switch už pomocí informací ví, kterým jeho portem sousedí s nějakým směrovačem:

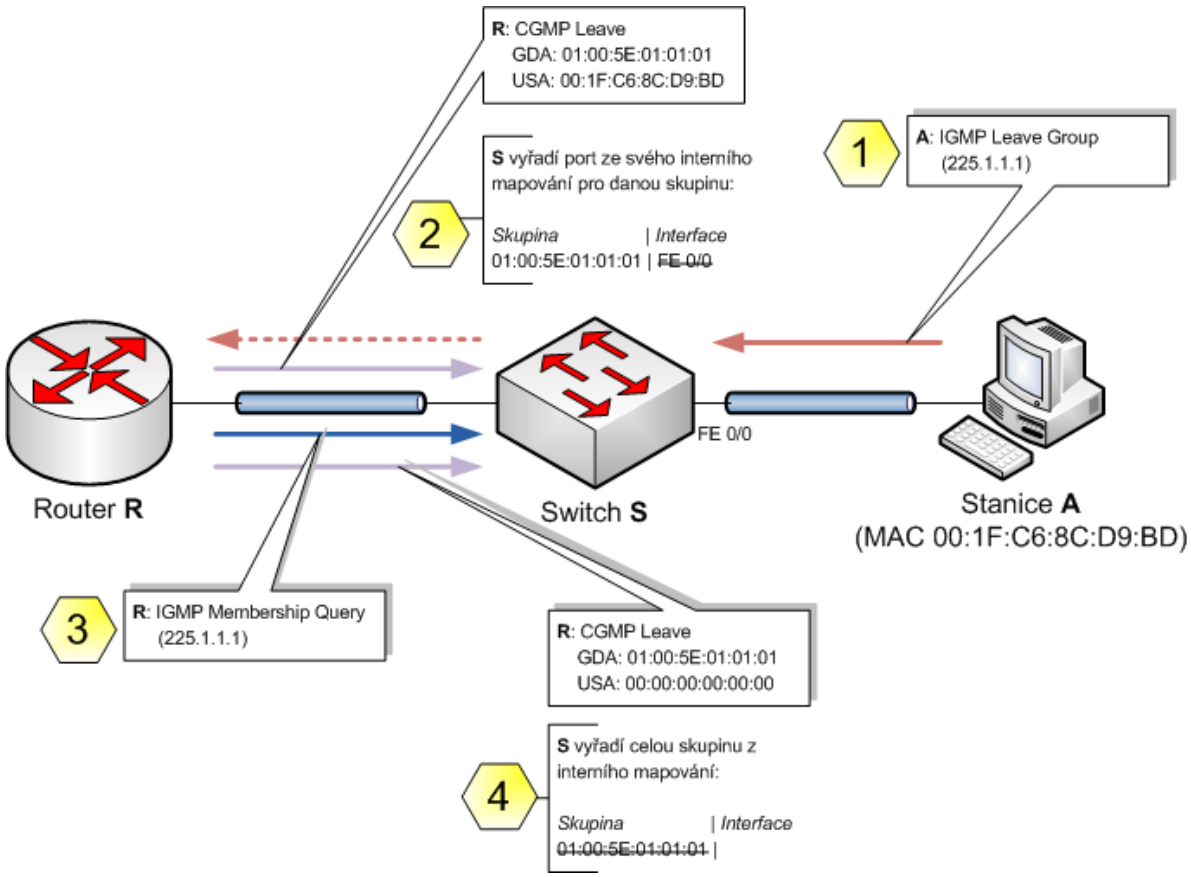
1) Stanice **A** se chce přihlásit k odběru dat z multicastingové skupiny 225.1.1.1, pošle tedy *IGMP Membership Report*, se kterým switch **S** nakládá jako z unicastem pro neznámého adresáta;
2) Router **R** uslyší zprávu od **A** a generuje *CGMP Join* (s parametry **#1** jak je uvádí Tabulka 4), který pošle **S**;
3) **S** zpracuje zprávu od **R** a podle jejího obsahu si do interního mapování přiřadí k požadované multicastové skupině port, na kterém se nachází **A**;



Obrázek 6 (CGMP při připojování k multicastu)

V případě odhlašování klienta platí scénář, který ilustruje Obrázek 7 a text:

1) **A** se rozhodne odhlásit od odběru dat z multicastové skupiny 225.1.1.1, proto pošle zprávu *IGMP Leave Group*;
2) **R** zpracuje zprávu od **A** a vyšle tedy *CGMP Leave* s kombinací parametrů **#2** přepínači **S**, aby vyřadil ze svého interního mapování port, na kterém se nacházela odhlašující se stanice;
3) **R** musí zkontrolovat zasláním *IGMP Membership Query*, jestli mu na daném síťovém segmentu zbyli ještě nějací případní odběratelé multicastingu pro skupinu 225.1.1.1;
4) Protože žádná další stanice odebírající multicast už není, a to jak na rozhraní FE 0/0, tak na zbývajících rozhraních **S**, tak žádná odpověď na výzvu nedorazí, a po chvíli **R** vyprší časovač čekání. Tím pádem zašle *CGMP Leave* se vstupními parametry **#5** díky kterému **S** zruší celé své interní mapování pro danou multicastingovou skupinu;



Obrázek 7 (CGMP při odhlašování od multicastu)

V současné době HW prostředky přepínačů dosáhly takové úrovně, že je standardně používana inspekce paketů tak, jak byla popsána v kapitole 3.2 IGMP snooping, tím pádem se proprietální protokoly jako CGMP staly historickým přežitkem.

3.4 Protocol Independent Multicast

Pro následující kapitoly a zejména protokoly, které se v nich popisují, mají většinou jakou nutnou podmínku své korektní funkčnosti spolupráci s PIM-SM. Začněme tedy nejprve jednoduchou rekapitulací faktů o směrovacích protokolech pro multicastingový provoz, abychom se pak propracovali až k popisu PIM-SM. Pro detailnější informace odkazuji čtenáře na různé oficiální materiály, jako například RFC 4601 [11] či RFC 3973 [12].

3.4.1 Distribuční stromy

Všechny směrovací protokoly pro multicasting (ať už PIM či DVMRP) používají ke svému fungování tzv. distribuční stromy – myšlenkovou abstraktní strukturu, která definuje cestu od zdroje až k příjemcům.

Tyto stromy můžeme rozdělit do dvou skupin:

- **source trees (zdrojové stromy)** – pro každý zdroj (kořen stromu) daného multicastingového provozu (vysílač v rámci konkrétní skupiny) je vybudováván samostatný strom nejkratších vzdáleností, ke všem jeho příjemcům;
- **shared trees (sdílené stromy)** – sázejí na nějaký ústřední bod v rámci síťové topologie, který sdružuje multicastingový provoz od více zdrojů či z více skupin a právě od něj je budován opět strom ke všem případným zájemcům.

Výhodou zdrojových stromů je, že pro daný vysílač obsahují vždy informace o nejkratší cestě ke všem svým příjemcům. Ovšem v případě, že je skupina rozsáhlá nebo obsahuje velké množství různých zdrojů, může být udržování všech zdrojových stromů značně náročné na výpočetní prostředky směrovačů.

Zmíněný nešvar řeší právě sdílené stromy, ve kterých slouží kořen (v terminologii často označován zkratkou RP = *rendezvous point*) jako prostředník mezi různými zdroji, které mu zasílají svá multicastová data, aby je přeposlal skupině příjemců. Výhodou tohoto topologického kompromisu v zasílání je, že udržování sdílených stromů je šetrnější k výkonu routerů, ovšem protipólem je, že tyto stromy obvykle neobsahují nejkratší možnou cestu od zdroje k příjemci.

Konvencí v označování zdrojových stromů je **(S, G)** u sdílených stromů pak **(*, G)**, kde **S** je adresa zdroje a **G** je adresa skupiny. Jak vidíme u sdílených stromů je S nahrazeno *, což označuje, že daný strom sdružuje provoz od několika různých zdrojů.

3.4.2 Multicastingové směrování

Hlavní rozdíl ve směrování mezi multicastem a unicastem je, že v případě unicastu se rozhodnutí o přeposlání činí na základě toho, kam mají být data doručena, zatímco v případě multicastu je směrování založeno na zdroji dat.

Přitom v případě multicastingu musí být směrovače schopné rozlišit, kterým „směrem“ leží zdroje dat a kterým pak příjemci, aby nedocházelo k smyčkám ve směrování. Klíčová je v tomto směru technologie RPF (*reverse path forwarding*), jejíž myšlenkou je: „Přepošli multicastová data na všechny své porty tehdy a jedině tehdy, když přišla portem, který *je ve směru* ke kořeni distribučního stromu.“ K tomu se používá porovnání zdrojové adresy paketu s unicastovou směrovací tabulkou routeru a právě od toho vzniklo i pojmenování PIMu, jakožto „protokolově nezávislého multicastingu“, který ke své činnosti využívá unicastové směrovací informace získané libovolným unicastovým směrovacím protokolem (OSPF, EIGRP, aj.).

Stejně jako můžeme unicastové směrovací protokoly rozdělit do dvou skupin (*distance vector* a *link-state*) podle logiky fungování, tak můžeme klasifikovat multicastingové směrovací protokoly do dvou skupin, a to podle módu, ve kterém pracují:

- **dense (hustý) mód** – pracuje inkluzívním přístupem, pravidelně zaplavuje celou síť multicastovým provozem, přičemž ty části stromu, na kterých nejsou žádní odběratelé, explicitně kořen upozorňují, aby jim nic nezasílal;
- **sparse (řídký) mód** – pracuje exkluzivním přístupem, strom je budován podle poptávky příjemců, kteří kořen explicitně upozorňují, že mají zájem odebírat data z nějaké multicastové skupiny.

Někdy se hovoří o tom, že dense mód je postaven na tzv. *push modelu*, zatímco sparse mód pracuje nad *pull modelem*.

Dense mód protokoly používají ke své činnosti zdrojových stromů a typickými představiteli jsou DVMRP či PIM-DM. Oproti tomu sparse mód protokoly operují většinou se sdílenými stromy nebo jsou schopny svou činnost účelově kombinovat mezi zdrojovými i sdílenými stromy a typickým představitelem je právě PIM-SM, který si popíšeme detailněji.

3.4.3 PIM – Sparse Mode

Jak bylo zmíněno obecně výše, PIM-SM tedy přeposílá multicastová data jen do oněch síťových segmentů, která o odběr výslovně požádala. K tomu je využíváno a vyžadováno administrativně konfigurovaného RP směrovače (jen jeden na každou šířenou multicastovou skupinu), který:

- sdružuje multicastové zdroje, jež se k němu registrují a zasílají mu jen jednu kopii dat;
- replikuje od zdrojů přijatá data a podle dané skupiny je distribuuje ke svým nahlášeným odběratelům.

V hierarchii PIM zařízení rozlišujeme ještě tzv. *DR = designated router*, který je prostředníkem za všechny PIM routery v rámci segmentu sítě či autonomního systému.

Protože je konvergence multicastové sítě značně komplexní, zavádí si PIM na pomoc stavový automat, pomocí něhož si určuje, v jaké fázi funkčnosti se router nachází, aby mohl patřičně reagovat na příjem řídících zpráv.

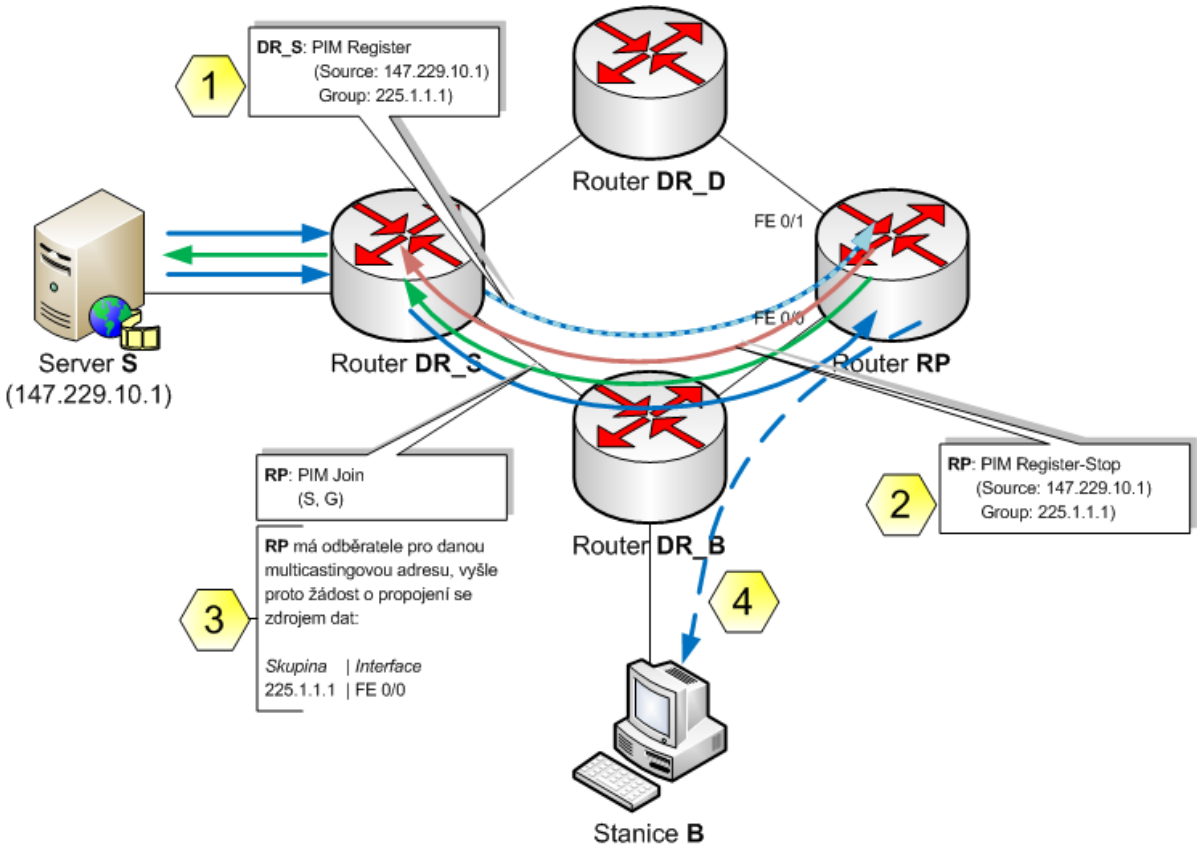
Následující tabulka shrnuje všechny druhy zpráv a jejich zevrubný význam v PIM:

Typ	Kód	Cílová adresa	Popis
Hello	0x0	224.0.0.13	Periodicky vysílán všemi PIM routery, obsahuje informace o multicastovým směrování a případně údaje ovlivňující případnou volbu DR na segmentu.
Register	0x1	RP	Unicastově ji vysílá DR, když se objeví nový zdroj multicastových dat, který by rád distribuoval obsah přes sdílený strom RP.
Register-Stop	0x2	Registrující	Unicastové vysílání RP k původci *PIM Register* zprávy, že došlo v pořádku k registraci zdroje a nyní může vysílat už klasicky.
Join/Prune	0x3	224.0.0.13	Tento typ zprávy slouží k rozšiřování, anebo ořezávání distribučního stromu, který směrovače zasílají RP či dalším routerů po cestě k němu či zdroji dat.
Bootstrap	0x4	224.0.0.13	Mechanismus, který umožňuje zautomatizovat šíření informace o RP v rámci domény, pomocí tzv. RP-setů a bootstrap routerů.
Assert	0x5	224.0.0.13	Řídící zpráva, kterou si mezi sebou PIM routery vyměňují, když dojde k detekci směrovací smyčky, aby opět stanovily integritu cesty a umožnili správnou funkci RPF.
Graft	0x6	Uzel rodiče ve stromu	Šíří se stromem směrem ke kořeni, stejný jako *PIM Join/Prune*, ale využívá ho ke své činnosti PIM-DM ve chvíli, kdy se objeví zájemce o odběr v již oříznuté větvi zdrojového stromu.
Graft-Ack	0x7	Uzel potomka ve stromu	Šíří se stromem směrem od kořene a potvrzuje opětovné napojení dané větve k odběru dat z multicastu.
Candidate-RP-Advertisement	0x8	BSR router	Kandidáti na RP dávají touto cestou hlavnímu BSR routeru vědet, pro které skupiny se mohou stát RP.

Tabulka 5 (PIM druhy zpráv)

Obrázek 8 a text ilustrují funkčnost PIM-SM ve chvíli, kdy se objeví nový zdroj multicastingových dat a následně se k jeho odběru přihlásí nová stanice, tento scénář předpokládá, že již na daných síťových segmentech byla provedena volba DR a tyto pak znají svůj RP:

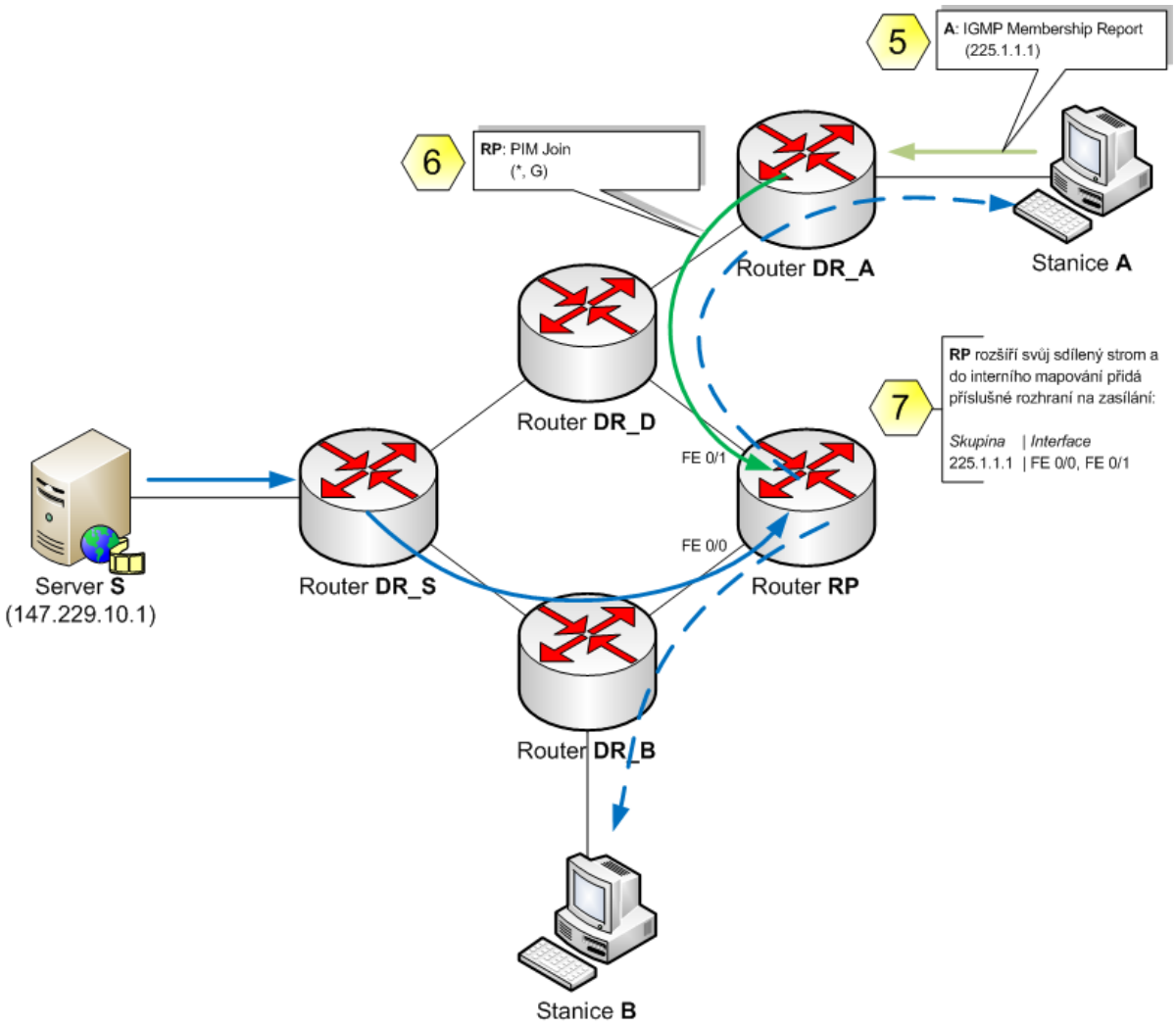
1) Server **S** je novým zdrojem multicastových dat, která začne ihned vysílat. Směrovač **DR_S** umístěný na stejném síťovém segmentu jako **S** odchytí jeho multicastový paket. Tento zpracuje a obalí ho vlastní *PIM Register* zprávou, kterou zašle unicastem **RP**, jenž je kořenem ve sdíleném stromě pro skupinu 225.1.1.1;
2) RP příjme zprávu *PIM Register* od **DR_S**, rozbalí ji a vyšle zprávu *PIM Register-Stop* s cílovou adresou **S**, kterou oznámí všem směrovačům po cestě k **S**, že mají přestat zasílat *PIM Register* zprávy, neb došlo k úspěšnému zaregistrování zdroje multicastových dat;
3) Ve chvíli, kdy **RP** detekuje nějaké odběratele dat (segment sítě ze stanicí **B**) pro 225.1.1.1 vyšle k **S** vlastní zprávu *PIM Join*, která ho s vysílačem multicastu propojí vlastním zdrojovým stromem (S, G) = (147.229.10.1, 225.1.1.1);
4) První multicastová data, která rozbalil ještě ze zprávy *PIM Register* od **DR_S** přepošle svým sdíleným stromem (*, 225.1.1.1), a protože už **DR_S** dostal příkaz, aby ukončil registrování (pomocí *PIM Register-Stop* od **RP**), tak všechna další data už putují jako klasický multicast;



Obrázek 8 (PIM-SM při připojování zdroje multicastu)

Následující body a Obrázek 9 ilustrují scénář připojování nového odběratele:

5) Stanice **A** se nyní přihlásí k odběru multicastových dat ze skupiny 225.1.1.1 pomocí zprávy *IGMP Membership Query*, kterou **DR_A** patřičně zpracuje;
6) Směrovač **DR_A** tedy vyšle *PIM Join* příkaz k **RP**;
7) **RP** díky zprávě *PIM Join* zjistí, že je potřeba rozšířit jeho sdílený strom pro šíření multicastu v požadované skupině a začne A přeposílat data od zdroje;



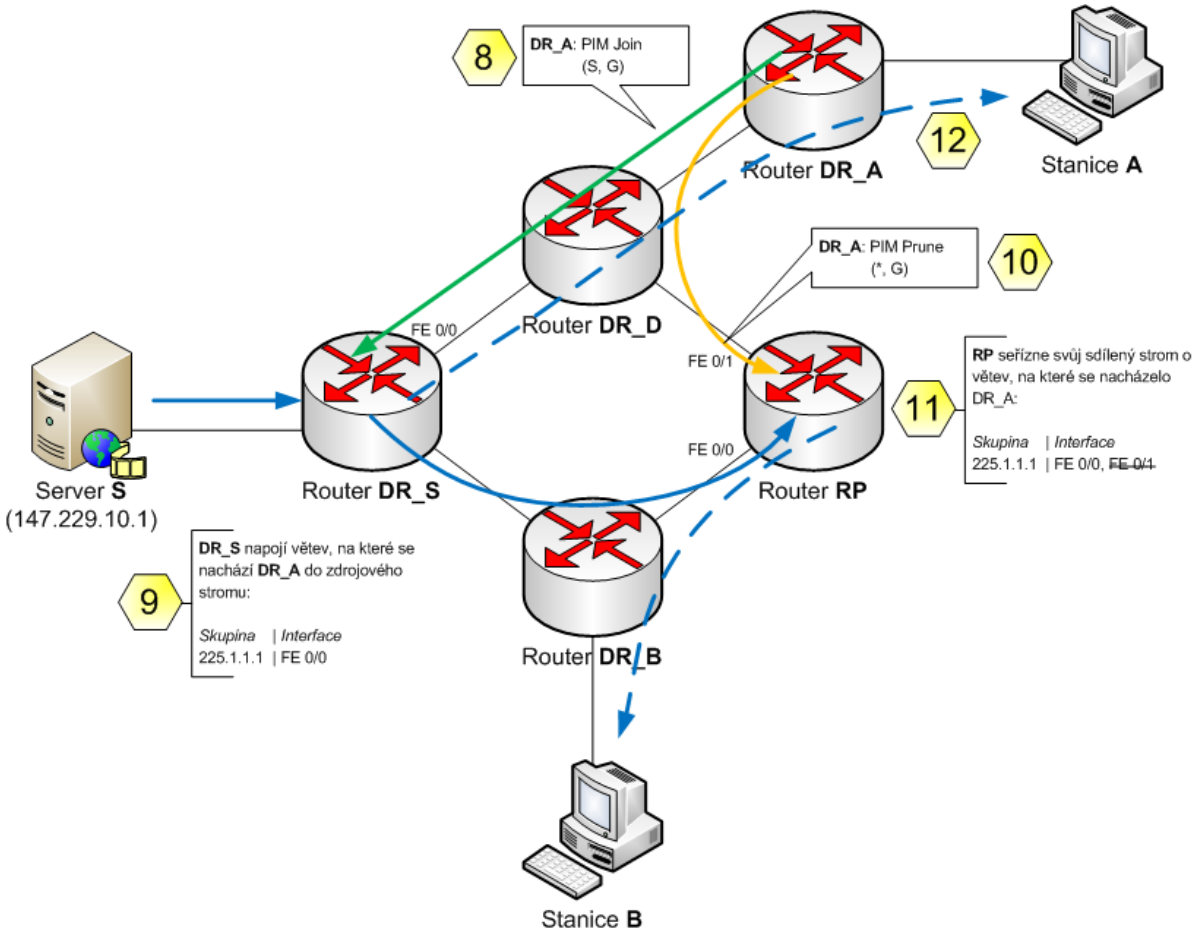
Obrázek 9 (PIM-SM při přidávání nového odběratele dat)

Je jasné, že RP routery jsou slabým místem v hierarchii v případě, kdyby došlo k výpadku nebo by objem multicastového provozu stoupl tak, že by bylo výhodnější odebírat data ze zdrojového stromu, než aby putovaly potenciálně delší cestou přes sdílený strom. A proto je i jednou z vlastností PIM-SM možnost sebeoptimalizace v případě, kdy to návrhář sítě uzná za vhodné. Když některý z routerů ve sdíleném stromě detekuje tuto skutečnost (typicky překročení velikosti datového provozu pomocí nějakého prahu), pokusí se o nalezení vlastní cesty ke zdroji pomocí zdrojového stromu,

který začne zpětně od sebe směrem ke zdroji budovat. Pokud narazí na zařízení, které už je napojeno zdrojovým stromem na zdroj multicastových dat, tak jednoduše k němu připojí svou větev.

V následujícím pokračování rozvineme naši ukázkovou topologii ještě o výše popisovaný problém:

8) **DR_A** detekuje, že **RP** nestíhá kvalitně dodávat multicastová data, proto se pokusí napojit na zdrojový strom s kořenem přímo v **S**, proto vyšle směrem k **S** zprávu *PIM Join*;
9) Ta se šíří až k **DR_S**, který je součástí zdrojového stromu (147.229.10.1, 225.1.1.1) a jenž zpracuje *PIM Join* tím, že nekratší větev k **DR_A** napojí do tohoto stromu;
10) Ve chvíli, kdy **DR_A** začíná odebírat data se zdrojového stromu místo ze sdíleného od **RP**, vyšle právě k **RP** zprávu *PIM Prune*;
11) **RP** po přijetí *PIM Prune*, odstřihne větev ze svého sdíleného stromu, na které se **DR_A** nachází, aby zbytečně nešířilo multicastová data tam, kde to už není potřeba;
12) Multicastová data ke stanici **A** nyní proudí zdrojovým stromem kratší cestou.



Obrázek 10 (PIM-SM při optimalizaci příjmu multicastu)

3.4.4 PIM – Bootstrap

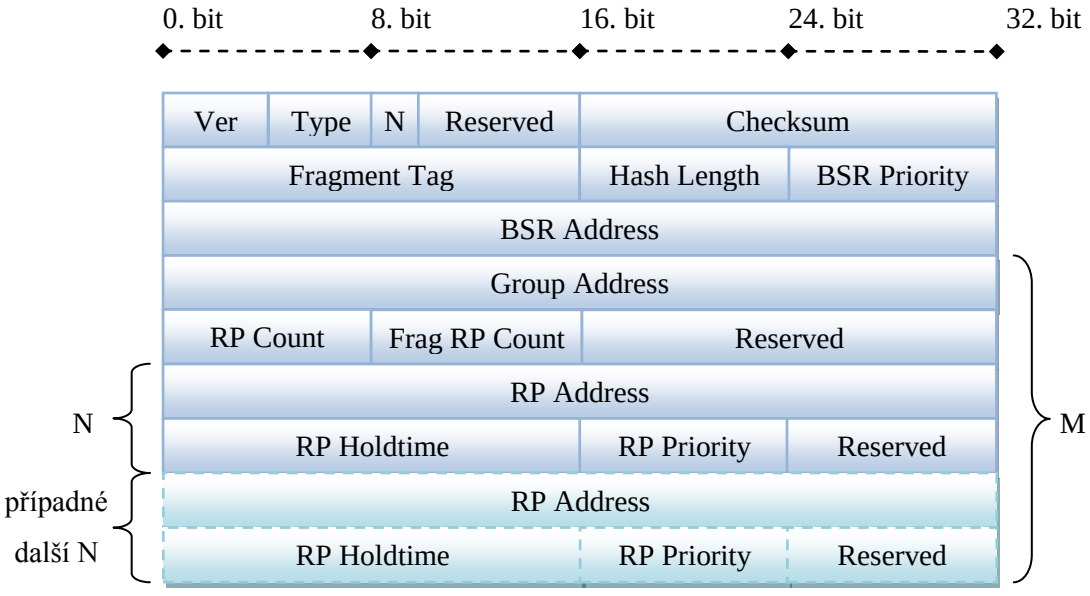
Jak už bylo napsáno, PIM-SM ke své korektní činnosti potřebuje znát RP pro každou aktivní multicastovou skupinu. Manuální nastavování takovéhoto mapování RP může být obzvláště ve velkých sítích náročná a úmorná práce. Někteří výrobci aktivních síťových prvků reagovali tvorbou vlastních proprietálních řešení, které by tuto činnost zautomatizovaly (např. Cisco Auto-RP [13]), což zavdalo podnět skupině IETF k standardizaci. Výsledkem jejich snažení je Bootstrapping rozšíření protokolu PIM od verze 2 popsané v RFC 5059 [14].

Bootstrap mechanismus v rámci jedné PIM domény vnímá několik rolí, pod kterými může směrovač vystupovat:

- **Candidate-RP (C-RP)** – routery s touto rolí ví o zdroji multicastu a můžou se tak stát RP pro danou skupinu;
- **Candidate-BSR (C-BSR)** – routery s touto rolí tvoří množinu kandidátů, ze které je na základě priority zvolen jeden BSR, což se dozví všechny PIM routery v dané doméně;
- **Boostrap Router (BSR)** – je vlastně moderátor multicastu v rámci domény, kterému se všechny C-RP ohlašují se svými kandidaturami na RP, z nich pak BSR vybírá určitou podmnožinu, kterou distribuuje všem PIM routerů jakožto výsledek mapování multicastové skupiny s RP.

Ke své činnosti využívá Boostrap mechanismus zprávy typu *PIM Bootstrap* a *PIM Candidate-RP-Advertisement*, tak jak uvádí Tabulka 5.

Zpráva *PIM Bootstrap* slouží ke komunikaci mezi všemi PIM routery. C-BSR router v ní udává svou prioritu pro elekci, BSR pomocí ní dává vědět o mapování RP ke skupinám a všechny PIM routery se pomocí ní dozvídají adresu zvoleného BSR. Obrázek 11 ukazuje strukturu paketu:



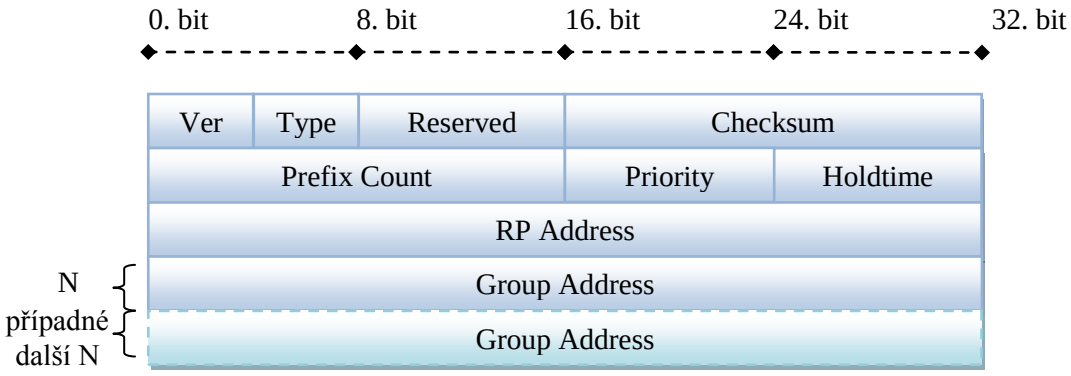
Obrázek 11 (struktura PIM Bootstrap)

- **Ver** – verze PIM protokolu, tento druh zprávy je podporován od verze 2;
- **Type** – nastaven na 0x4;
- **N** – pokud nastaven na 1, tak router po přijetí už nepřeposílá tuto zprávu dál;
- **Fragment Tag** – náhodně generované číslo pro každou kompletní *PIM Bootstrap* zprávu, která je díky svému velkému obsahu často fragmentována na více části, jednotlivé fragmenty pak mají shodý Fragment Tag;
- **Hash Length** – hraje roli v případě, kdy je pro jednu skupinu více kandidátů C-RP;
- **BSR Priority** – priorita C-BSR v rozsahu 0 až 255 (čím větší, tím lepší);
- **BSR Address** – adresa BSR pro celou doménu;
- **Group Address** – Multicastová adresa, pro kterou je níže uvedena soupiska jejích RP;
- **RP Count** – počet RP pro inzerovanou multicastovou skupinu;
- **Frag RP Count** – počet RP pro danou multicastovou skupinu v tomto fragmentu;
- **RP Address** – adresa konkrétního RP;
- **RP Holdtime** – doba ve vteřinách, po kterou je RP pro inzerovanou skupinu ještě platný;
- **RP Priority** – priorita, se kterou je RP nabízen (čím nižší, tím lepší);

Obrázek 11 ukazuje, že zpráva *PIM Bootstrap* obsahuje tedy aspoň jednu inzerovanou skupinu (označeno svorkou „M“), přičemž tato skupina obsahuje jeden či více RP (označených svorkou „N“).

Zprávou *PIM Candidate-RP-Advertisement* oznamují C-RP své kandidatury na RP pro různé multicastové skupiny routeru BSR, přičemž ten pak z nich vybírá určitou podmnožinu (RP-set), kterou šíří všem PIM routerů v doméně.

Strukturu *PIM Candidate-RP-Advertisement* ilustruje Obrázek 12 a významy jednotlivých políček jsou analogické s políčky ve zprávě *PIM Bootstrap*, až na **Prefix Count**, které obsahuje počet **Group Address**, ke kterým se směrovač hlásí jako C-RP:



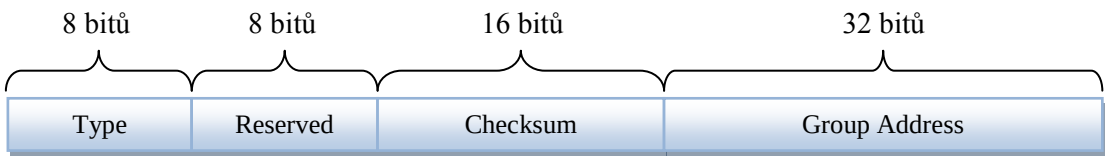
Obrázek 12 (strukutra PIM Candidate-RP-Advertisement)

3.5 Router-port Group Management Protocol

RGMP je protokol druhé vrstvy, který umožňuje směrovačům komunikovat s přepínači a ovlivňovat tok multicastového provozu těmito zařízeními úpravou jejich CAM tabulek. Idea za tímto protokolem je: „IGMP snooping či CGMP pomáhá přepínačům v efektivní kontrole nad porty, kterými procházejí multicastová data nahlášeným klientům, ovšem nepomáhá přepínačům v obdobné optimalizaci nad porty, kterými protéká provoz určený směrovačům“.

Primárně je tento protokol a jeho nasazení určeno do prostředí páteřních sítí, aby se odlehčilo vytížení routerů od provozu, který jim není určen (pro danou skupinu nemají nahlášeny žádné odběratele). I když je tento protokol proprietálním výtvorem Cisco Systems Inc., je k dispozici jeho informační koncept v podobě RFC 3488 [15] a další užitečné informace může čtenář najít v související odborné literatuře od Cisco Press – např. [16]. Stejně jako v případě CGMP je postaven na jednosměrné komunikaci routerů vůči switchům; jakýkoli paket RGMP, který by router přijal, je automaticky ignorován a zahozen.

Struktura RGMP paketu:



Obrázek 13 (struktura RGMP zpráv)

- **Type** – určuje druh zprávy (*Hello* – 0xFF, *Bye* – 0xFE, *Join* – 0xFD, *Leave* – 0xFC);
- **Reserved** – vždy nastaven na 0x0;
- **Checksum** – kontrolní součet IP paketu, sestavován stejnou metodou jako v případě IGMP;
- **Group Address** – IPv4 adresa multicastové skupiny, které se zpráva týká.

RGMP jsou zabalovány do RGMP rámců, které přepínač rozliší od ostatního provozu pomocí konstantní cílové adresy 01:00:5E:00:00:19. Tabulka 6 shrnuje všechny druhy zpráv a jejich zevrubný význam.

Typ	Kód	GA	Popis
Hello	0xFF	0.0.0.0	Po příjmu tohoto paketu přestane switch na daném portu zasílat jakýkoli multicastový provoz
Bye	0xFE		Po přijetí této zprávy se navrátí daný port na přepínači do původního stavu, takže přeposílá veškerý multicast i tímto rozhraním
Join	0xFD	IPv4 multicast	Portem se povolí zasílání multicastu specifikované skupiny
Leave	0xFC		Portem se zakáže zasílání multicastu specifikované skupiny

Tabulka 6 (RGMP druhy zpráv)

RGMP funguje v podmíněné součinnosti se směrovacím protokolem PIM-SM. Tabulka 7 ukazuje, kdy směrovač generuje jednotlivé RGMP zprávy a jakým způsobem s nimi přepínač nakládá:

Typ	Podmínky zasílání routerem	Způsob zpracování switchem
Hello	Ve chvíli, jak je RGMP zapnuto, dává router switchy najevo, že nehodlá přijímat žádná multicastová data od všech skupin, pokud si je explicitně nevyžádá. *RGMP Hello* se zasílají se stejnou periodicitou jako *PIM Hello* zprávy (obvykle 30 sekund).	Když switch na daném portu zachytí tuto zprávu, označí port jako RGMP aktivní a přestane na něj zasílat multicastová data ze všech skupin.
Bye	Ve chvíli, kdy se podpora RGMP na routeru vypne, dá se vědět všem podřízeným switchům, aby obnovily obecně úplné zasílání multicastu tomuto zařízení.	Switch po přijetí zprávy označí daný port jako RGMP pasivní a navrátí ho do interního mapování multicastového provozu pro všechny existující skupiny.
Join	Ve chvíli, kdy router vytvoří zprávu *PIM Join*, zašle spolu s ní na daném rozhraní i zprávu *RGMP Join*.	Přidá port, na kterém byla zpráva přijata, do interního mapování k zasílání multicastového provozu určeného skupině uvedené ve zprávě.
Leave	Ve chvíli, kdy router vytvoří zprávu *PIM Prune*, zkontroluje, jestli se dané rozhraní nachází případně v dalších interních mapováních pro jiné multicastové skupiny. Pokud se daný port nenachází, zašle ještě zprávu *RGMP Leave*.	Odebere port, na kterém byla zpráva přijata, z interního mapování k zasílání multicastového provozu určeného skupině uvedené ve zprávě.

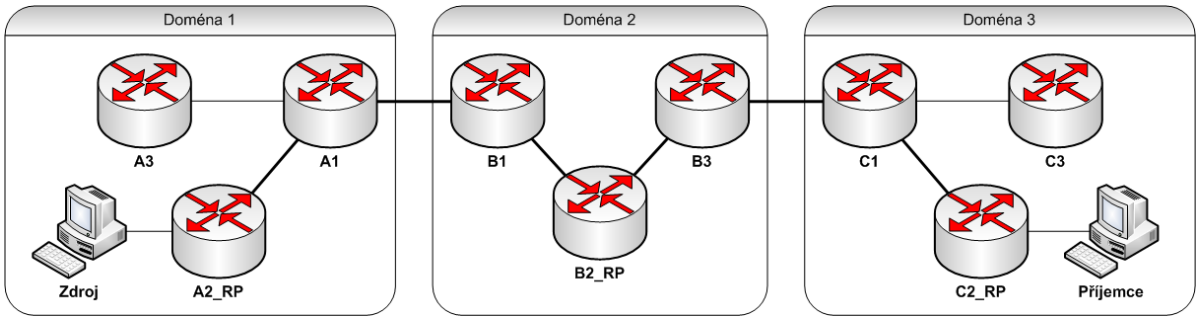
Tabulka 7 (chování RGMP)

3.6 Multicast Source Discovery Protocol

MSDP standardizované pro obecné použití dokumentem RFC 3618 [17] je periodický protokol na transportní vrstvě určený pro IPv4, který dovoluje v součinnosti s PIM-SM si vyměňovat informace mezi různými autonomními systémy o tamní multicastové struktuře. V rámci MSDP jsou vyměňovány informace o lokálních zdrojích multicastových dat a routery díky tomu vědí o případných RP v sousedících doménách. MSDP architektura se pak vyznačuje následujícími vlastnostmi:

- odolnost v nezávislosti – každá PIM-SM doména má pokud možno vlastní RP směrovač;
- optimalizace ve znalosti – díky informacím od sousedů je umožněna existence domén, ve kterých jsou jen příjemci multicastingu, aniž by musela daná doména šířit informace o příslušnosti k multicastové skupině do celého Internetu;

V rámci protokolu MSDP mezi sebou komunikují RP propojených domén, kteří jsou spolu buď přímo, nebo nepřímo propojeni sérií MSDP zařízení spárovaných spojením (= *MSDP peering*). Přičemž rozlišujeme různé MSDP peery typu, všechny pak zobrazuje Obrázek 14:



Obrázek 14 (kategorie MSDP peerů)

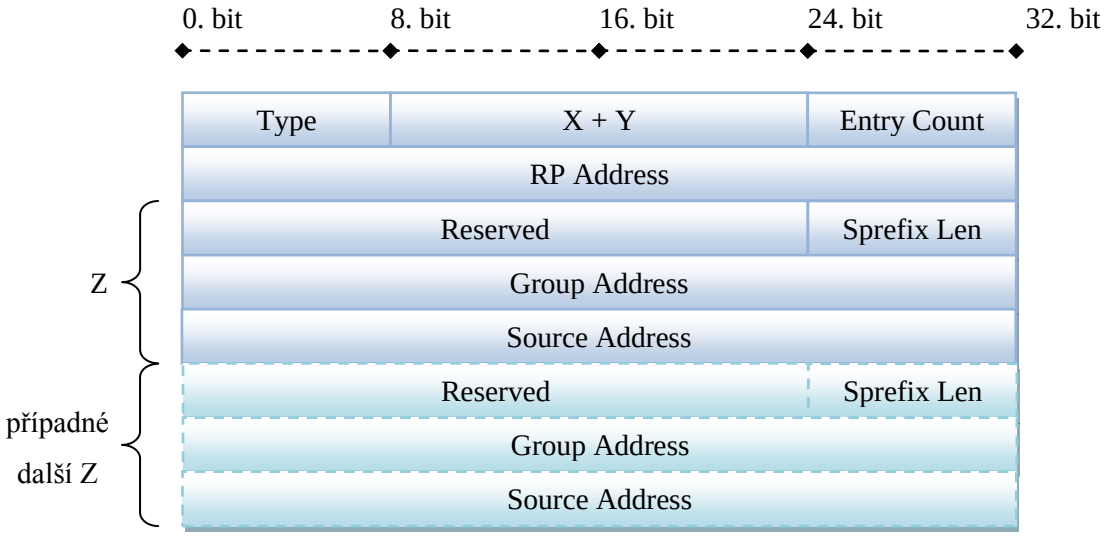
- **RP MSDP peer** – v rámci dané domény je směrovač lokálním RP pro skupinu, tyto MSDP peery jsme schopni rozdělit ještě do podkategorií:
  - **Source-side** – RP ležící v doméně, ve které se nachází zdroj multicastových dat, existuje vždy jen jeden MSDP peer pro každou skupinu (směrovač **A2_RP**);
  - **Receiver-side** – RP ležící nejblíže příjemci multicastových dat, typicky jich existuje více (směrovač **C2_RP**);
  - **Intermediate** – jakékoli RP nacházející se po cestě mezi Source-side RP a Receiver-side RP (směrovač **B2_RP**), počet závislý na topologii;
- **Non-RP MSDP peer** – všechny ostatní směrovače, které se podílejí na MSDP komunikaci mezi RP (směrovače **A1**, **B1**, **B3**, **C1**).

MSDP rozlišuje následující typy zpráv, přičemž používá standardní TLV zapouzdřující mechanismus (typ T určuje sémantiku obsahu V, který má délku L), jejichž výčet udává Tabulka 8:

Typ	Kód	Popis
Source-Active	0x1	Tato zpráva se používá k oznamování aktivních zdrojů multicast v dané doméně, taktéž se používá k přenosu úplně prvních multicastových dat z nově registrovaného zdroje.
Source-Active Request	0x2	Zpráva se používá k vyžádání seznamu aktivních zdrojů v dané doméně pro specifikovanou multicastovou skupinu.
Source-Active Response	0x3	Odpovědní zpráva s aktivními zdroji v dané doméně na původní *MSDP Source-Active Request*.
KeepAlive	0x4	Tato zpráva je periodicky zasílána každých 60 vteřin ke kontrole MSDP peeringu. Pokud od přijetí poslední zprávy uplyne více než 75 vteřin je MSDP peering mezi dvěma zařízeními zrušen.

Tabulka 8 (MSDP typy zpráv)

Maximální velikost libovolné MSDP zprávy je 9192 oktetů bez hlaviček zapouzdřujících protokolů. Obrázek 15 ukazuje strukturu segmentu zprávy *MSDP Source-Active*:



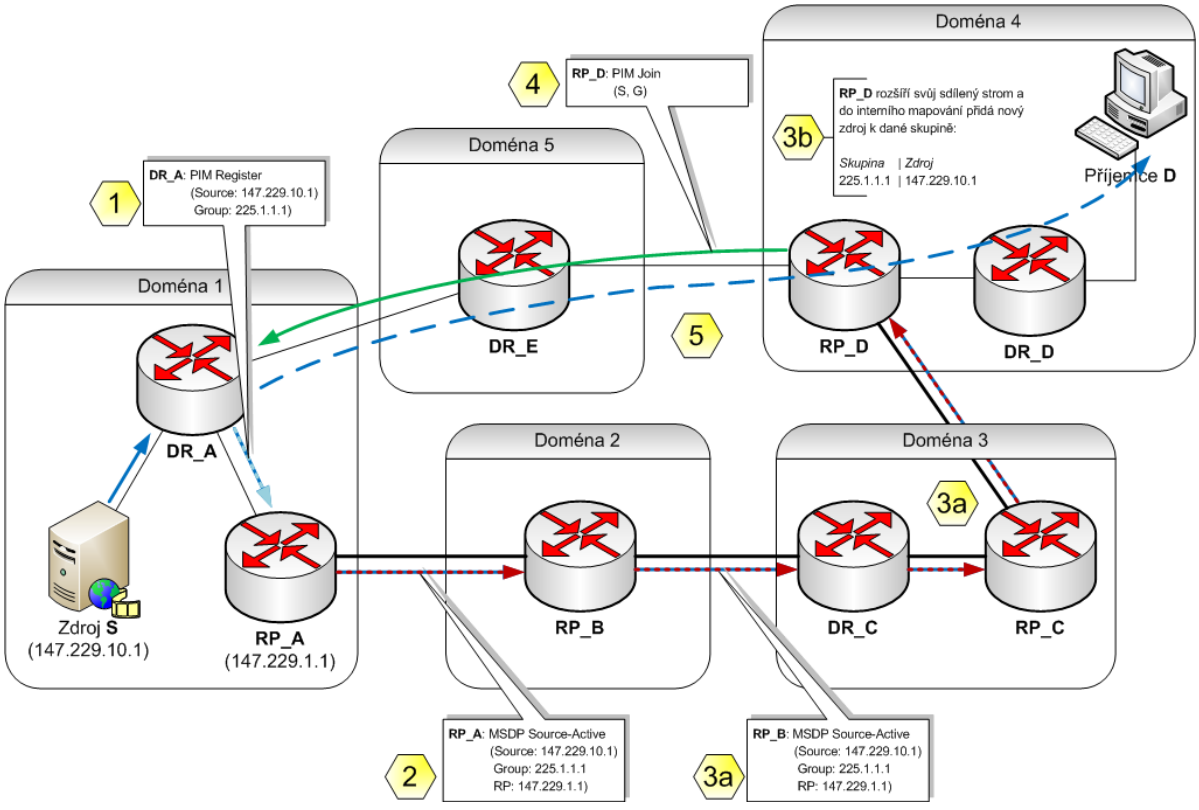
Obrázek 15 (struktura MSDP Source-Active zprávy)

- **Type** – určuje druh MSDP zprávy, v tomto případě nastaven na 0x1;
- **X** – délka kontrolní části zprávy v oktetech = 8 + 12 × **Entry Count**;
- **Y** – jestliže nejsou zabalena žádná multicastová data, tak je nastaveno na 0x0, v opačném případě odpovídá délce zapouzdřených dat v oktetech;
- **Entry Count** – počet **Z** částí;
- **RP Address** – adresa RP v doméně, ve které jsou uvedené zdroje multicastingu aktivní;
- **Reserved** – vždy nastaveno na 0x0 a ignorováno zařízením zpracovávajícím zprávu;
- **Sprefix Len** – maska zdroje adresy, vždy nastavena na 32 (0x20);
- **Group Address** – multicastová adresa skupiny, do které zdroj zasílá svá data;
- **Source Address** – IP adresa zdroje;

Následující Obrázek 16 a komentář ilustrují fungování MSDP v případě, kdy se objeví nový zdroj multicastingu a informace o jeho připojení je rozšířena napříč doménami:

1) Zdroj **S** začne vysílat multicastová data pro skupinu 225.1.1.1, tento první paket je odchycen směrovačem **DR_A**, který ho vezme a zabalí a pošle uvnitř zprávy *PIM Register* příslušnému RP pro Doménu 1, což je router **RP_A**;
2) **RP_A** příjme zprávu *PIM Register* a jen co rozbalí její multicastový obsah, tak ho hned přepouzdří do vlastní zprávy *MSDP Source-Active*, kterou je jakožto source-side RP povinen vytvořit a zaslat svým MSDP peerům.

3) Tato zpráva putuje MSDP strukturou, přičemž každý router po cestě provede nejprve RPF (kvůli případným smyčkám) a následně porovná její obsah se svou multicastovou strukturou, jmenovitě se zaregistrovanými skupinami;
    a) Pokud nenajde shodu, tak přeposílá zprávu vlastním MSDP peerům (případy routerů, které jsou buď Non-RP jako **DR_C**, a nebo intermediate RP, jimiž jsou **RP_B** a **RP_C**);
    b) Pokud najde shodu (inzerovaný zdroj dat odpovídá skupině, ke které jsou v rámci domény registrovaní nějací odběratelé), což je případ receiver-side **DR_D**, tak rozbalí multicastový obsah zprávy *MSDP Source-Active* a pošle ji svým sdíleným stromem všem příjemcům (*, G);
4) Zároveň s tím se pokusí napojit na nejkratší zdrojový strom (S, G) tím, že vyšle zprávu *PIM Join*, kterou se postupně prořeže až k **S** přes router **DR_E**;
5) Multicastová data od **S** nakonec proudí vzniklým zdrojovým stromem k **DR_D** a od něj pak ke stanicím registrovaným k odběru skupiny 225.1.1.1, což je např. počítač **D**.



Obrázek 16 (MSDP při připojování nového zdroje multicastu)

3.7 Multiprotocol BGP

M-BGP s nejnovější revizí v RFC 4760 [18] je rozšířením klasického unicastového směrovacího protokolu BGP (RFC 4721 [19] – primárně určeného pro výměnu směrovacích dat mezi samostatnými autonomními systémy, typicky poskytovateli internetového připojení) o podporu multicastingu a VPN v prostředí IPv4. Díky tomu je schopen propojovat multicastové struktury napříč celým Internetem.

Původní protokol BGP je navržen tak univerzálně, že rozšíření o skupinové směrování znamenalo jen zavedení nových netranzitivních atributů MP_REACH_NLRI a MP_UNREACH_NLRI. Obecně tyto atributy tak upozorňují směrovač, pro jaké sítě může či nemůže šířit informace o dalším aktivním bodě (*next hop point*) v závislosti na zpracování příslušným síťovým protokolem.

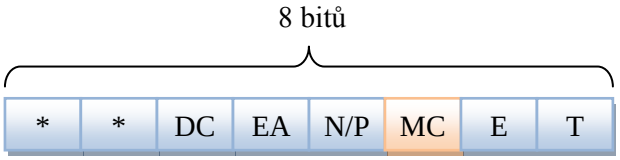
V případě multicastingu M-BGP pracuje v součinnosti s PIMem, do kterého jsou distribuované informace injektovány a on má pak možnost si přepočítávat své zdrojové a sdílené stromy podle aktuální situace. Multicastovým next-hop směrovačem je tak obvykle chápán rendezvous point ležící v jiném autonomním systému nebo přímo zdroj multicastingových dat.

3.8 Multicast Extension to OSPF

Jak již název kapitoly napovídá, jedná se o rozšíření unicastového směrovacího protokolu OSPF (nejaktuálněji v RFC 2328 [20]) o podporu multicastingu. Definované je v RFC 1584 [21] a dost neaktuální informace o jeho nasazení a interoperabilitě v rámci jiných multicastingových protokolů lze najít v souvisejících kompilacích nejlepších praktik RFC 2715 [22] a RFC 1585 [23].

MOSPF rozšiřuje původní sadu OSPF LSA (*link state advertisement*) o nový typ, a to *Group-Membership-LSA*, pomocí kterých směrovače zjišťují, na jakých síťových segmentech jsou aktivní příslušné multicastové skupiny. Informace o multicastu jsou tímto způsobem jednoduše zapracovány do pro tento problém univerzální stavové databáze OSPF, ve které se dá cesta od zdroje multicastových dat (kořene) vybudovat jednoduše pomocí SPT (*shortest path tree*) stromu.

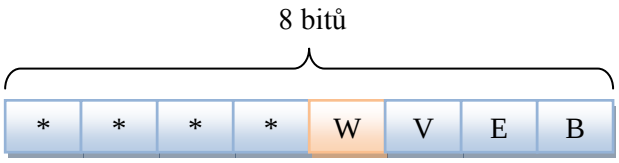
Změny, ke kterým došlo, jsou strukturálního rázu v paketech. Routery bez MOSPF jsou při komunikaci s případnými MOSPF-routery zcela kompatibilní, jednoduše nové parametry ignorují. Nejprve bylo třeba upravit *OSPF Hello* pakety, aby si mezi sebou daly směrovače najevo, že nově podporují i multicasting, a to úpravou pole **Options** (240. až 247. bit těla této zprávy):



Obrázek 17 (struktura pole Options v OSPF Hello zprávě)

- **DC** – nakládání s vyžádanými okruhy;
- **EA** – nakládání se LSA obsahujícími externí atributy;
- **N/P** – chování v případě not-so-stubby area LSA;
- **MC** – podpora rozšíření MOSPF;
- **E** – šíření LSA mezi rozdílnými autonomními systémy v stub area;
- **T** – funkce TOS;

Další úpravy prodělal *OSPF Router-LSA*, který směrovač vysílá každým svým portem a jenž určuje stav a schopnosti daného rozhraní v rámci autonomního systému. Tělu této zprávy bylo pozměněno pole **rtype** (160. až 167. bit):

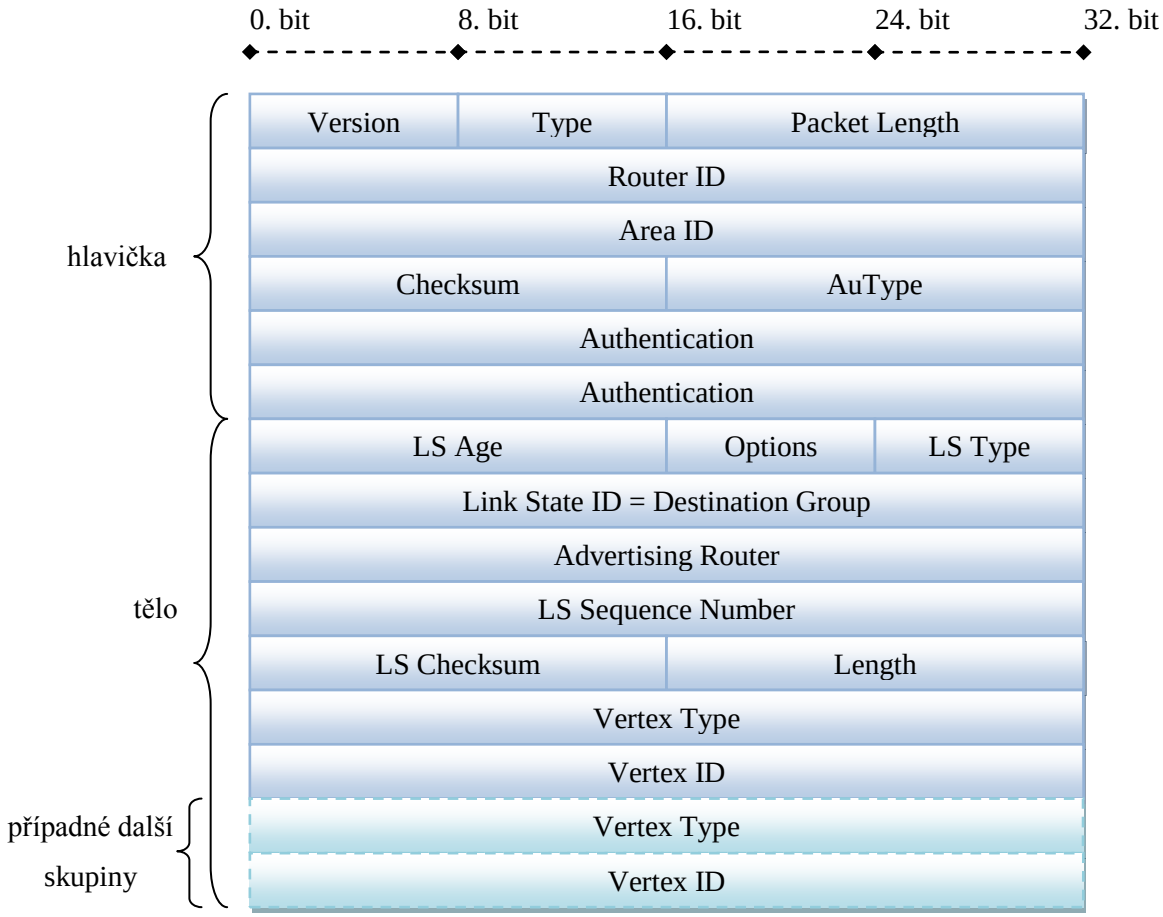


Obrázek 18 (struktura pole rtype v OSPF Router-LSA)

- **W** – označuje, že router na daném rozhrání je příjemcem všeho multicastového provoz bez rozdílu;
- **V** – označuje, že je router koncovým bodem virtuálního propojení, užívá se v případě tranzitních oblastí;
- **E** – označuje, že je router hraniční pro daný autonomní systém;
- **B** – označuje, že je router hraniční pro danou oblast;

A na závěr byl zaveden zcela nový druh LSA, a to *MOSPF Group-Membership-LSA*, který je specifický pro každou OSPF oblast. Směrovač vyšle toto LSA pokaždé, když se v rámci nějakého síťového segmentu přihlásí stanice k odběru multicastových dat (např. pomocí *IGMP Membership*) nebo když je router hraniční pro danou autonomní oblast a šíří informaci o tom, že v oblasti existují odběratelé multicastu pro dané skupiny.

Struktura *MOSPF Group-Membership-LSA* je uvedena níže, pro úplnost popisuje Obrázek 19 formát celého OSPF paketu (hlavička + tělo):



Obrázek 19 (struktura OSPF Group-Membership-LSA paketu)

- **Type** – druh OSPF paketu, v případě LSA nastaveno na 0x3;
- **Packet Length** – délka paketu v bytech;
- **Router ID** – v rámci oblasti unikátní 32bitový identifikátor;
- **Area ID** – identifikátor oblasti, ke které se daný paket váže, u virtuálních spojení v případě použití tranzitivních oblastní nastaveno na 0.0.0.0;
- **Checksum** – klasický IP kontrolní součet;
- **AuType** – druh použité autentizace paketu (žádná = 0x0, plaintext heslo = 0x1, MD5 = 0x2);
- **Authentication** – při použití jiné než nezabezpečené informace nese patřičné parametry;
- **LS Age** – čas v sekundách od doby úplného vytvoření LSA;
- **Options** – viz výše;
- **LS Type** – druh LSA, v případě *MOSPF Group-Membership-LSA* nastaveno na 0x6;
- **Link State ID** – IPv4 adresa multicastové skupiny;
- **Advertising Router** – router ID směrovače, který LSA inicioval;

- **LS Sequence Number** – pořadové číslo sloužící ke kontrole na duplikáty či příliš staré LSA;
- **LS Checksum** – Fletcherův kontrolní součet [24] těla zprávy mimo položku „LS Age“;
- **Length** – stejné jako **Packet Length**;
- **Vertex Type** – určuje druh vrcholu ve stromu SPT;
- **Vertex ID** – obsahuje router ID, které jednoznačně ohodnocuje daný vrchol ve stromě;

MOSPF je první implementací skupinového směrování, ovšem v návaznosti na strukturu typických multicastových dat se od jeho používání v komerční sféře brzy upustilo, a to zejména díky jeho velké režii a nárokům na výpočetní prostředky routerů. V současnosti je toto rozšíření OSPF velkými výrobci aktivních síťových prvků (Cisco Systems či Juniper Networks[3]) nepodporováno, neb už existují vhodnější proprietální řešení.

[3] *Juniper Networks* - http://www.juniper.net/

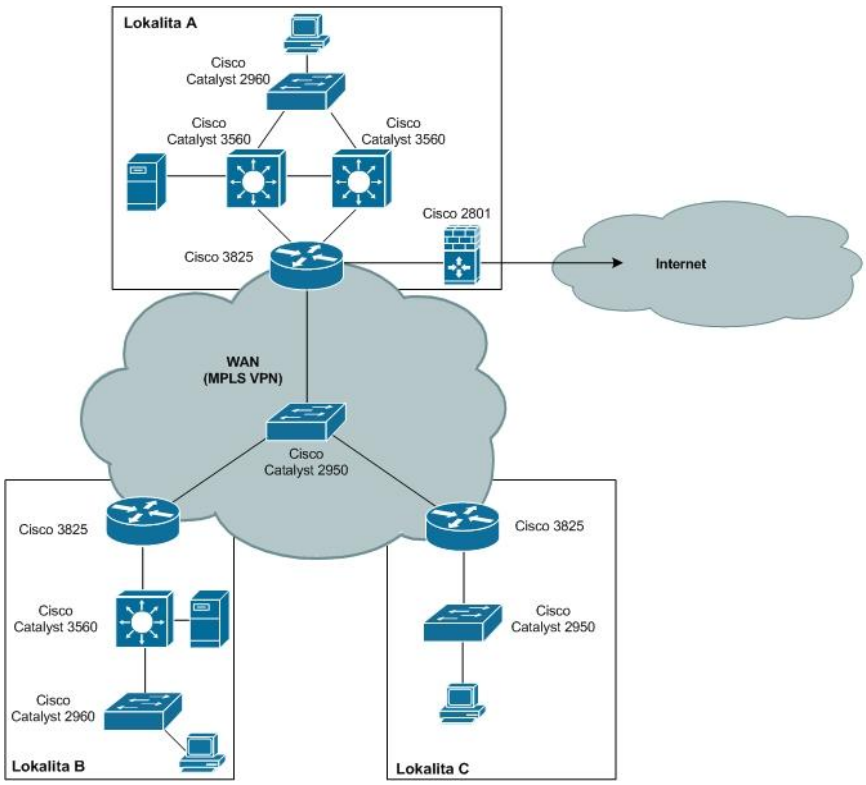
4 Praktická realizace

V rámci společnosti ANECT a.s.[4] byl vypsán následující interní projekt, jehož cílem je nasazení technologie IP Multicast v síťovém prostředí.

Modelová společnost uvažuje o zpřístupnění multimediálního obsahu každému ze svých zaměstnanců, přičemž zdroje tohoto obsahu budou jednak interní (dedikovaný server), a jednak je zvažována i možnost přebírání multimediálního obsahu ze zdrojů umístěných v Internetu. Kritériem je co nejefektivnější šíření tohoto obsahu v síti společnosti tak, aby byl doručován jen k aktivním příjemcům a jeho vliv na jiný síťový provoz byl pokud možno co nejmenší. Společnost je tvořena třemi pobočkami, které jsou vzájemně propojeny pronajatými digitálními okruhy (topologie hvězda) s implementovanou MPLS VPN technologií. Každá z poboček disponuje přepínanou LAN sítí postavenou na Ethernet-přepínačích Cisco řady Catalyst 2950, 2960 a 3560. Směrovače (PE a CE routery) jsou z řad Cisco 2800 a 3800. V úvahu by měly být brány i bezpečnostní aspekty (možnosti filtrace či omezení registrace k nevhodným zdrojům).

4.1 Úvodní specifikace

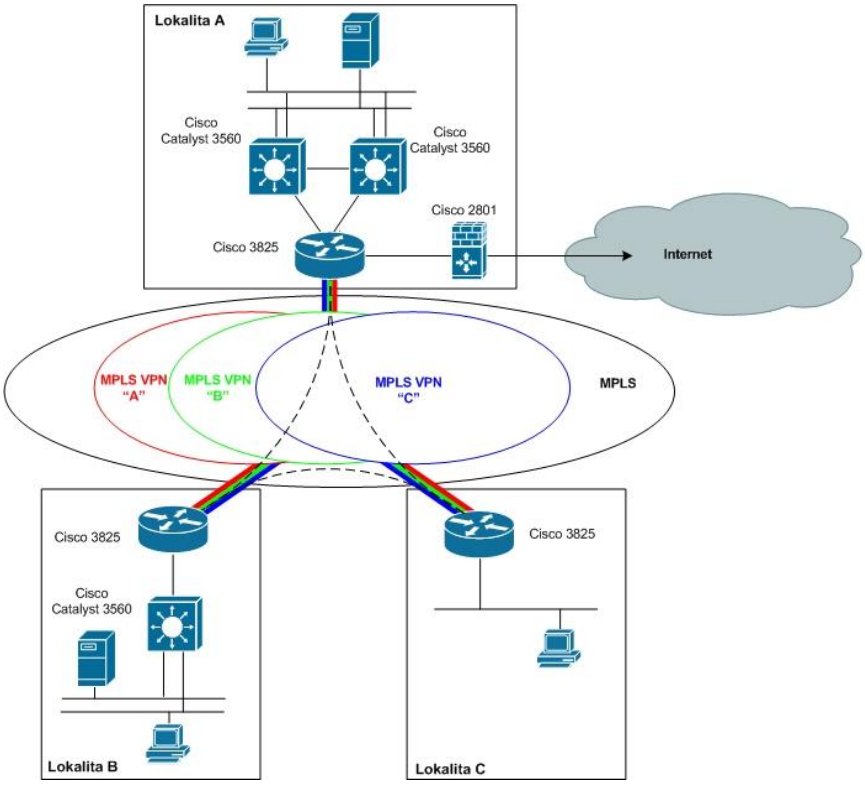
4.1.1 L2 topologie



Obrázek 20 (návrh linkové vrstvy)

[4] *ANECT a.s.* - http://www.anect.com

4.1.2 L3 topologie



Obrázek 21 (návrh síťové vrstvy)

4.1.3 Popis architektury

Navržená architektura kopíruje požadavky zadání do tří autonomních lokalit propojených na L2 vrstvě topologií typu hvězda, která je na L3 zapouzdřena třemi MPLS VPN spojnicemi s rozběhnutým M-BGP. Uvnitř každé z lokalit poběží samostatný OSPF směrovací protokol. Z lokality A je vyvedena spojnice chráněná jednoduchým HW firewallem do Internetu. L3 switche v téže lokalitě budou mít mezi sebou rozběhnutý HSRP [25] linkový protokol k poskytnutí redundance. Vše bude tvořit jedinou PIM-SDM doménu, kde bude staticky konfigurován RP na směrovač v lokalitě A.

Předpokládaná aktivní síťová zařízení:

- 3× směrovač Cisco 3825;
- 1× směrovač Cisco 2801;
- 3× přepínač Cisco Catalyst 3560;
- 2× přepínač Cisco Catalyst 2960;
- 2× přepínač Cisco Catalyst 2950;

4.2 Výsledné řešení

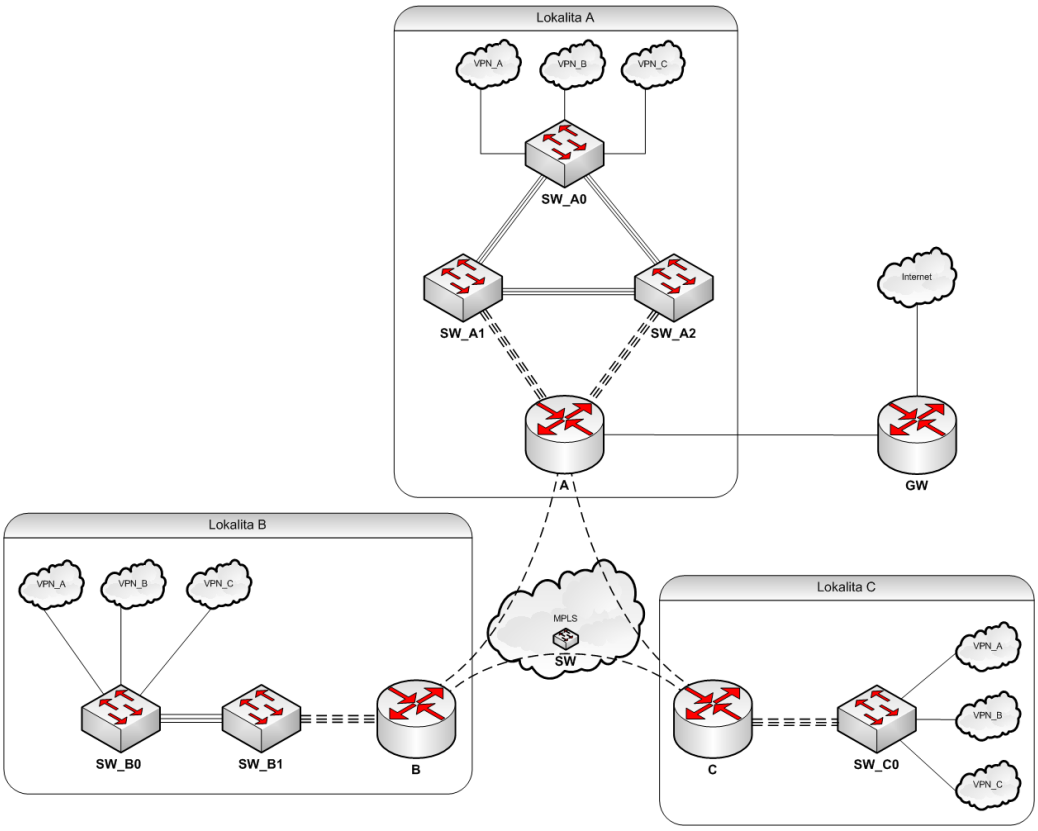
Následující podkapitola se zabývá úplným popisem laboratoře, která byla sestavena, aby co nejvěrněji modelovala a naplnila požadavky kladené úvodní specifikací.

4.2.1 Použitá zařízení

Hostname	Platforma	IOS
GW	Cisco 2801 Integrated Services Router	c2801-advipservicesk9-mz.124-9.T7.bin
A	Cisco 3845 Integrated Services Router	c3845-spservicesk9-mz.124-15.T8.bin
B	Cisco 3825 Integrated Services Router	c3825-spservicesk9-mz.124-15.T8.bin
C	Cisco 2801 Integrated Services Router	c2801-advipservicesk9-mz.124-11.T1.bin
SW	Cisco Catalyst 2960 Series Switches	c2960-lanbase-mz.122-50.SE1.bin
SW_A0	Cisco Catalyst 2950 Series Switches	c2950-i6k2l2q4-mz.121-22.EA13.bin
SW_A1	Cisco Catalyst 3560 Series Switches	c3560-advipservicesk9-mz.122-46.SE.bin
SW_A2	Cisco Catalyst 3560 Series Switches	c3560-advipservicesk9-mz.122-46.SE.bin
SW_B0	Cisco Catalyst 2950 Series Switches	c2950-i6k2l2q4-mz.121-22.EA13.bin
SW_B1	Cisco Catalyst 3560 Series Switches	c3560-advipservicesk9-mz.122-46.SE.bin
SW_C0	Cisco Catalyst 2960 Series Switches	c2960-lanbase-mz.122-50.SE1.bin

Tabulka 9 (použitá zařízení)

4.2.2 Adresní plán



Obrázek 22 (obecný pohled)

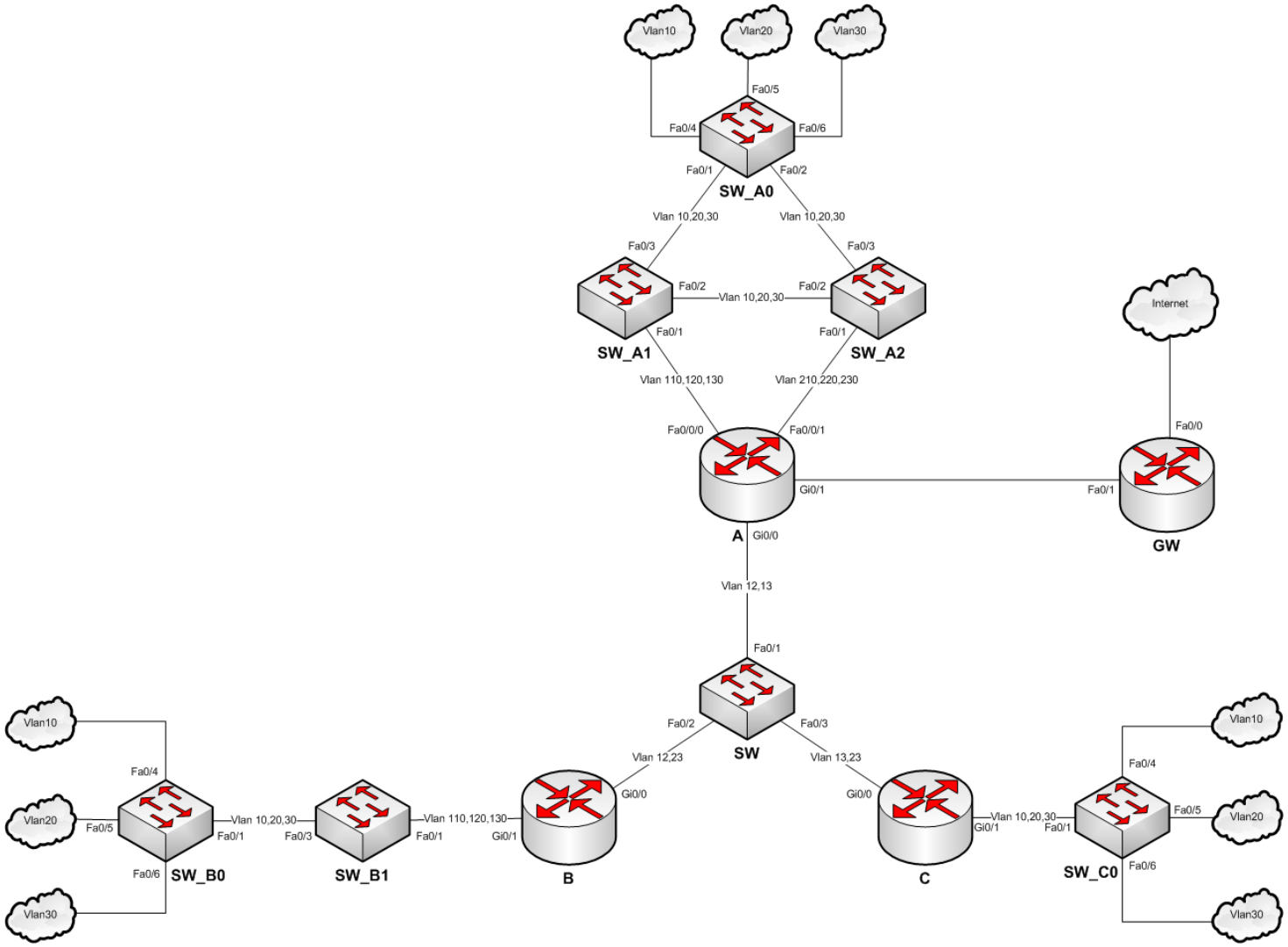
Jak ukazuje Obrázek 22, celá výsledná realizace pracuje se třemi MPLS VPN (nazvanými VPN_A, VPN_B, VPN_C) v každé z lokalit (pojmenovaných A, B, C). V rámci Cisco zařízení tedy tři různé

směrovací VRF instance. Navíc je zde oddělený adresní prostor mezi jednotlivými směrovači pro ustavení BGP-peeringu. O speciální VRF jménem VPN_OVERLAY a jejím významu se lze více dočíst v podkapitole 4.2.6 Přístup k Internetu. Orientaci v použitých adresách poskytuje Tabulka 10:

Adresa/Maska	VRF	Popis
192.168.1.1/32		Loopback adresa routeru **A**
192.168.2.2/32		Loopback adresa routeru **B**
192.168.3.3/32		Loopback adresa routeru **C**
192.168.12.0/30		Adresní prostor virtuálního subinterfaceu mezi routery **A** a **B**
192.168.13.0/30		Adresní prostor virtuálního subinterfaceu mezi routery **A** a **C**
192.168.23.0/30		Adresní prostor virtuálního subinterfaceu mezi routery **B** a **C**
172.16.0.0/24	VPN_OVERLAY	Adresní prostor spojnice mezi routery **A** a **GW**, sloužící jako adresa výchozí brány pro všechny lokality a jejich VPN, tato pak prosakuje i do ostatních VRF instancí
10.1.1.1/32	VPN_A	Loopback adresa routeru **A** pro VPN_A
10.1.110.0/30	VPN_A	Adresní prostor spojnice virtuálního subinterfaceu routeru **A** a virtuálního Vlan110 rozhraní switche **SW_A1** pro VPN_A
10.1.210.0/30	VPN_A	Adresní prostor spojnice virtuálního subinterfaceu routeru **A** a virtuálního Vlan210 rozhraní switche **SW_A2** pro VPN_A
10.1.10.0/24	VPN_A	Interní adresní prostor klientů v lokalitě A pro VPN_A
10.2.1.1/32	VPN_A	Loopback adresa routeru **B** pro VPN_A
10.2.110.0/30	VPN_A	Adresní prostor spojnice virtuálního subinterfaceu routeru **B** a virtuálního Vlan110 rozhraní switche **SW_B1** pro VPN_A
10.2.10.0/24	VPN_A	Interní adresní prostor klientů v lokalitě B pro VPN_A
10.3.1.1/32	VPN_A	Loopback adresa routeru **C** pro VPN_A
10.3.10.0/24	VPN_A	Interní adresní prostor klientů v lokalitě C pro VPN_A
10.1.2.1/32	VPN_B	Loopback adresa routeru **A** pro VPN_B
10.1.120.0/30	VPN_B	Adresní prostor spojnice virtuálního subinterfaceu routeru **A** a virtuálního Vlan120 rozhraní switche **SW_A1** pro VPN_B
10.1.220.0/30	VPN_B	Adresní prostor spojnice virtuálního subinterfaceu routeru **A** a virtuálního Vlan220 rozhraní switche **SW_A2** pro VPN_B
10.1.20.0/24	VPN_B	Interní adresní prostor klientů v lokalitě A pro VPN_B
10.2.2.1/32	VPN_B	Loopback adresa routeru **B** pro VPN_B
10.2.120.0/30	VPN_B	Adresní prostor spojnice virtuálního subinterfaceu routeru **B** a virtuálního Vlan120 rozhraní switche **SW_B1** pro VPN_B
10.2.20.0/24	VPN_B	Interní adresní prostor klientů v lokalitě B pro VPN_B
10.3.2.1/32	VPN_B	Loopback adresa routeru **C** pro VPN_B
10.3.20.0/24	VPN_B	Interní adresní prostor klientů v lokalitě C pro VPN_B
10.1.130.0/30	VPN_C	Adresní prostor spojnice virtuálního subinterfaceu routeru **A** a virtuálního Vlan130 rozhraní switche **SW_A1** pro VPN_C
10.1.230.0/30	VPN_C	Adresní prostor spojnice virtuálního subinterfaceu routeru **A** a virtuálního Vlan230 rozhraní switche **SW_A2** pro VPN_C
10.1.30.0/24	VPN_C	Interní adresní prostor klientů v lokalitě A pro VPN_C
10.2.130.0/30	VPN_C	Adresní prostor spojnice virtuálního subinterfaceu routeru **B** a virtuálního Vlan130 rozhraní switche **SW_B1** pro VPN_C
10.2.30.0/24	VPN_C	Interní adresní prostor klientů v lokalitě B pro VPN_C
10.3.30.0/24	VPN_C	Interní adresní prostor klientů v lokalitě C pro VPN_C
239.0.0.10/32		Adresa multicastového distribučního stromu pro VPN_A
239.0.0.20/32		Adresa multicastového distribučního stromu pro VPN_B
239.0.0.30/32		Adresa multicastového distribučního stromu pro VPN_C

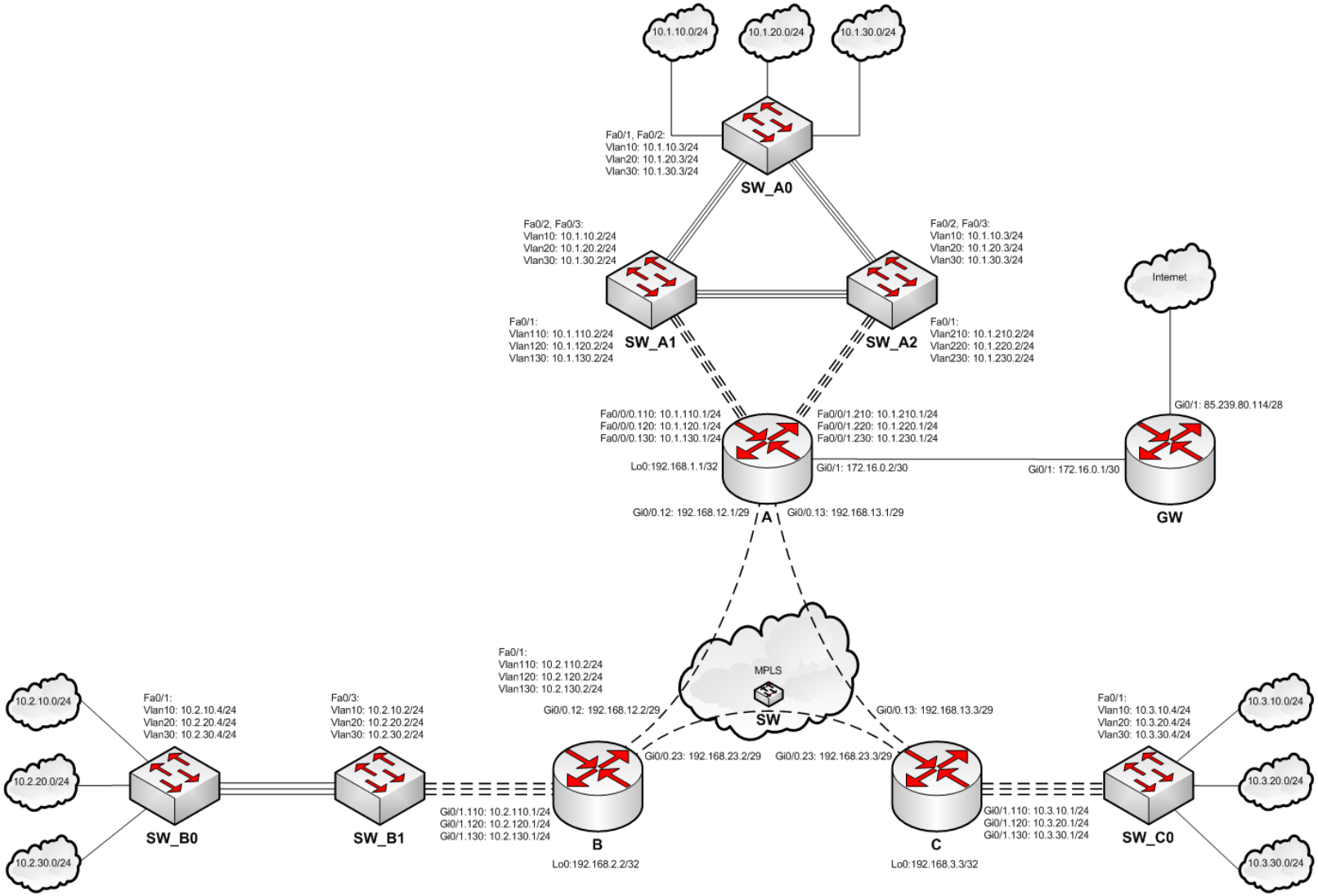
Tabulka 10 (přehled použitých IPv4 adres)

4.2.3 L2 topologie



Obrázek 23 (linková vrstva – fyzické zapojení a VLAN)

4.2.4 L3 topologie



Obrázek 24 (síťová vrstva – adresování)

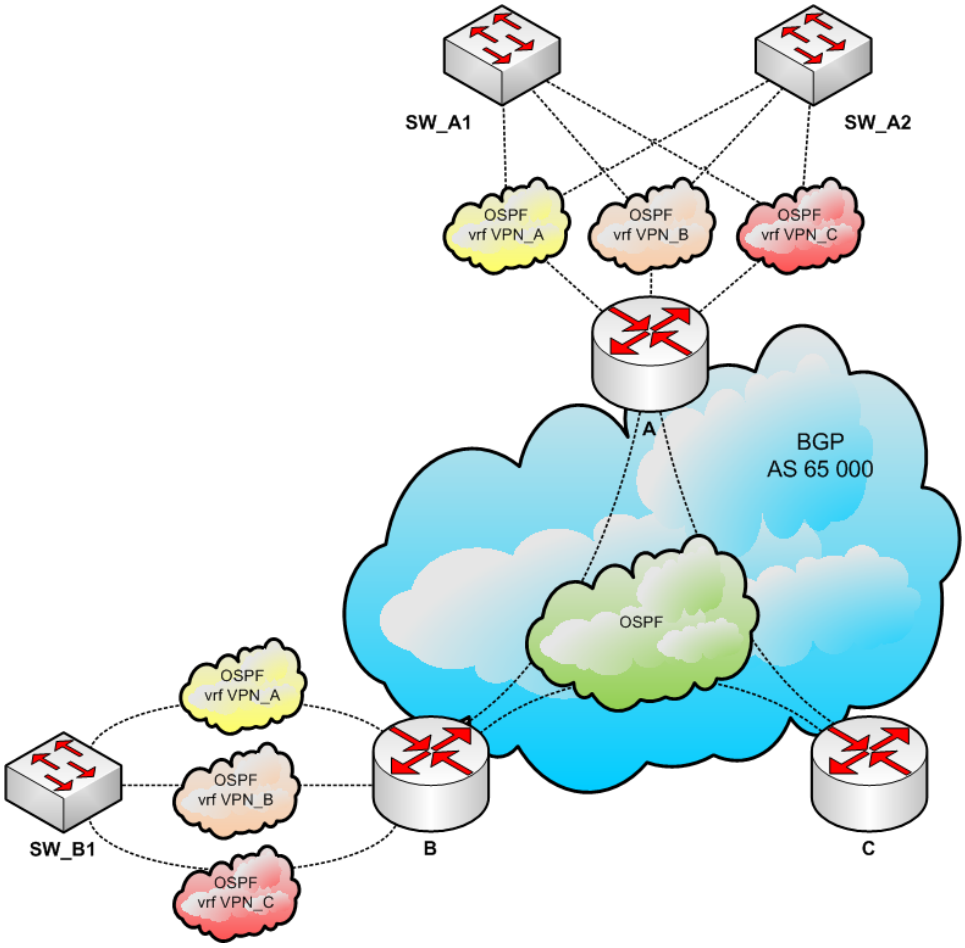
4.2.5 Směrování v rámci lokalit

V rámci páteřní sítě mezi lokalitami byl spuštěn jeden OSPF proces, který pokrývá adresní prostory virtuálních subinterfaceů, kterými jsou routery **A**, **B** a **C** vzájemně propojeny, a prostor těch loopbackových adres jednotlivých routerů, které nejsou součástí žádné VRF. Obrázek 25 ho uvádí zakreslený zeleně.

V každé lokalitě je pro každou instanci VRF spuštěn samostatný OSPF proces, jakožto vnitřní směrovací protokol (IGP – *interior gateway protocol*). Na obrázku níže se jedná o žluté, oranžové a červené mráčky.

Protože jsou všechny lokality mezi sebou propojeny MPLS VPN sítí, byl zvolen protokol BGP, jakožto vnější směrovací protokol (EGP – *exterior gateway protocol*) pro předávání směrovacích tabulek mezi jednotlivými lokalitami (v obrázku vyznačen modře). Bylo zvoleno číslo 65 000 pro identifikaci autonomního systému, ve kterém se všechny lokality nacházejí.

Mezi BGP a IGP OSPF procesy probíhá vzájemná redistribuce směrovacích informací. IGP OSPF do BGP distribuuje informace o sítích připojených v dané lokalitě (v případě směrovače **C** se v návaznosti na postavenou topologii provádí redistribuce přímo připojených rozhraní) a opačně je z BGP do těchto IGP OSPF procesů předávána znalost o připojených sítích v jiných lokalitách.



Obrázek 25 (rozvržení směrovacích protokolů)

Uveďme si výňatek z konfigurace routeru **B** pro představu, co vše je potřeba udělat, aby bylo výše popsaných cílů směrování dosáhnout, a to pro VPN_A (pro zbývající VPN_B a VPN_C je konfigurace analogická):

Konfigurace	Komentář k příkazu
ip vrf VPN_A	!samostatná VRF instance
rd 65000:10	!označovač cest
route-target export 65000:10	!export/import označených
route-target import 65000:10	cest v rámci VRF_A
route-target import 65000:40	!nasátí def.gw z VRF_OVERLAY
router ospf 10 vrf VPN_A	!OSPF v rámci VRF
log-adjacency-changes	
capability vrf-lite	!zakázání reverzních kontrol
redistribute bgp 65000 subnets	!redistribuce z BGP
network 10.2.0.0 0.0.255.255 area 0	!síť, do které se šíří
default-information originate	!router zná výchozí bránu
router ospf 1	!OSPF na páteři
log-adjacency-changes	
network 192.168.0.0 0.0.255.255 area 0	!páteřní síť
router bgp 65000	!BGP pro komunikaci
no synchronization	s lokalitami
bgp log-neighbor-changes	
neighbor 192.168.1.1 remote-as 65000	!BGP-peering se sousedem
neighbor 192.168.1.1 update-source Loopback0	!zdrojové rozhraní
neighbor 192.168.3.3 remote-as 65000	
neighbor 192.168.3.3 update-source Loopback0	
no auto-summary	!classless směrová tabulka
address-family vpnv4	!multi VPN podpora do BGP
neighbor 192.168.1.1 activate	
neighbor 192.168.1.1 send-community both	
neighbor 192.168.3.3 activate	
neighbor 192.168.3.3 send-community both	
exit-address-family	
address-family ipv4 vrf VPN_A	!VPN_A směrové informace
redistribute ospf 10 vrf VPN_A route-map OSPF2BGP	!redistribuce z OSPF pomocí
no synchronization	route-mapy
exit-address-family	
access-list 1 permit 10.2.0.0 0.0.255.255	!redistribuje pouze lokálně
route-map OSPF2BGP permit 1	připojené VPN_* sítě
match ip address 1	

Tabulka 11 (konfigurace routeru B pro routing)

4.2.6 Přístup k Internetu

Celá sestavená topologie pak byla připojena do Internetu, a to přes spojnici na routeru **GW**, kde byl spuštěn NAT N:1 pro překlad vnitřních privátních adres na veřejnou adresu 85.239.80.114 k tomuto projektu speciálně přidělenou.

Ještě je nutné zmínit způsob distribuce cesty k výchozí bráně ze systému, kterou je dle topologie spojnice mezi směrovači **A** a **GW**. Tato spojnice je součástí speciální VRF instance nazvané VPN_OVERLAY. Do této VRF jsou importovány směrovací informace ze všech zbylých VRF, tedy VPN_A, VPN_B a VPN_C, aby měla kompletní přehled o připojené topologii. Naopak z VPN_OVERLAY je pomocí jednoduché route-mapy distribuována jen statická cesta na výchozí bránu systému (router **GW**). Vše ilustruje výňatek z konfigurace na směrovači **A**:

Konfigurace	Komentář k příkazu
`ip vrf VPN_OVERLAY`	`!samostatná VRF instance`
` rd 65000:0`	
` route-target import 65000:10`	`!naimportuje informace`
` route-target import 65000:20`	`z ostatních VRF instancí`
` route-target import 65000:30`	
` export map DEFAULT`	`!exportuje jen určité cesty`
` route-target export 65000:0`	`definované route-mapou níže`
`interface GigabitEthernet0/1`	
` ip vrf forwarding VPN_OVERLAY`	`!zařazení rozhraní do VRF`
` ip address 172.16.0.2 255.255.255.252`	
`router bgp 65000`	
` address-family ipv4 vrf VPN_OVERLAY`	`!zařazení VPN_OVERLAY do BGP`
`  redistribute static`	
`  default-information originate`	`!router zná výchozí bránu a`
` exit-address-family`	`nachází se v této VRF`
` address-family ipv4 vrf VPN_C`	
`  default-information originate`	`!router zná výchozí bránu`
` exit-address-family`	`pro VPN_C`
` address-family ipv4 vrf VPN_B`	
`  default-information originate`	`!router zná výchozí bránu`
` exit-address-family`	`pro VPN_B`
` address-family ipv4 vrf VPN_A`	
`  default-information originate`	`!router zná výchozí bránu`
` exit-address-family`	`pro VPN_A`
`ip route vrf VPN_OVERLAY 0.0.0.0 0.0.0.0 172.16.0.1`	`!statické směrování na GW`
`ip access-list standard DF-GW permit 0.0.0.0`	`!povolovací ACL`
`route-map DEFAULT permit 1`	`!pomocí této route mapy je`
` description === Oznacovac na def. GW spojeni ===`	`označena novým tagem`
` match ip address DF-GW`	`statická cesta k GW, kterou`
` set extcommunity rt  65000:40 additive`	`si ostatní VRF naimportují`

Tabulka 12 (konfigurace redistribuce adresy výchozí brány na routeru A)

4.2.7 PIM

Pro distribuci multicastingového směrování byl dle požadavků zadání zvolen protokol PIM s tím, že byly vyzkoušeny (a níže shrnuty) dva jeho módy činnosti.

Prvním je PIM-SDM (*PIM Sparse-Dense Mode*), přičemž tento mód činnosti není nijak standardizován a je proprietálně dostupný na zařízeních firmy Cisco. Ve zkratce umožňuje dynamicky přepínat v rámci nakonfigurovaného rozhraní mezi oběma klasickými módy činnosti PIMu podle toho, zda je (použije PIM-SM) či není (použije PIM-DM) dostupná informace o RP pro multicastovou skupinu. Díky této flexibilitě výrazně snižuje výpočetní nároky na provoz PIMu ve smíšených distribučních systémech. Druhým je pak klasický PIM-SM s rozšířením o Bootstrapping, který je standardizovanou nástavbou PIMv2 a jenž přidává automatizované šíření informací o RP mezi mroutery.

Ať už se jedná o první či druhý mód činnosti, bylo nutné provést následující kroky k tomu, aby se PIM šířil vždy nejen v rámci VRF instance v jedné lokalitě, ale i mezi lokalitami samotnými:

1) zapnout multicastové směrování globálně i pro jednotlivé VRF instance;
2) povolit PIM na rozhraní, které slouží jako zdrojové pro BGP-peering;
3) nakonfigurovat MDT (*multicast distribution tree*);
4) povolit PIM na páteřních rozhraních;
5) povolit PIM na všech VRF rozhraních;

Uveďme si nyní patřičně zkrácenou ukázkovou konfigurace z routeru **B** s uvedením, kterého bodu je dosahováno daným příkazem (ukázka je pro VPN_A, zbývající analogicky):

Konfigurace	Bod
`ip vrf VPN_A`	3)
`mdt default 239.0.0.10`	
`ip multicast-routing`	1)
`ip multicast-routing vrf VPN_A`	1)
`interface Loopback0`	2)
`ip pim sparse-dense-mode`	
`interface GigabitEthernet0/0.12`	4)
`ip pim sparse-dense-mode`	
`interface GigabitEthernet0/0.13`	4)
`ip pim sparse-dense-mode`	
`interface Loopback1`	5)
`ip pim sparse-dense-mode`	
`interface GigabitEthernet0/1`	5)
`ip pim sparse-dense-mode`	
`interface FastEthernet0/0/0.110`	5)
`ip pim sparse-dense-mode`	

Tabulka 13 (konfigurace routeru A pro šíření multicastových stromů v multiVRF systému)

5 Testování

5.1 Test č. 0 – Konektivita

Před samotným testováním multicastingových struktur si zkontrolujeme na několika příkladech konektivitu v rámci celé topologie.

Shrňme tedy základní předpoklady do několika bodů:

1) Propojení směrovačů **A**, **B** a **C** na páteřní síti, tvořící jádro MPLS systému;
2) Klienti v různých lokalitách, ale ve stejné VPN mohou vzájemně komunikovat;
3) Klienti ležící v různých VPN nemohou spolu komunikovat;
4) Všichni klienti bez rozdílu se jsou schopni dostat do sítě Internet přes spojnici ve VPN_OVERLAY na směrovači **A** a překlad adres NAT na směrovači **GW**;

Bod 1) zkontrolujeme výpisem z globální směrovací tabulky na páteřních routerech, což je ilustrováno na výstupu z **A**:

```
A#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     192.168.12.0/29 is subnetted, 1 subnets
C       192.168.12.0 is directly connected, GigabitEthernet0/0.12
     192.168.13.0/29 is subnetted, 1 subnets
C       192.168.13.0 is directly connected, GigabitEthernet0/0.13
     192.168.23.0/29 is subnetted, 1 subnets
O       192.168.23.0 [110/2] via 192.168.13.3, 1w2d, GigabitEthernet0/0.13
                     [110/2] via 192.168.12.2, 1w4d, GigabitEthernet0/0.12
     192.168.1.0/32 is subnetted, 1 subnets
C       192.168.1.1 is directly connected, Loopback0
     192.168.2.0/32 is subnetted, 1 subnets
O       192.168.2.2 [110/2] via 192.168.12.2, 1w4d, GigabitEthernet0/0.12
     192.168.3.0/32 is subnetted, 1 subnets
O       192.168.3.3 [110/2] via 192.168.13.3, 1w2d, GigabitEthernet0/0.13
```

Jak vidíme, směrovač **A** je přímo propojen se směrovači **B** a **C** (cesty označeny jako „C“), je si také díky protokolu OSPF běžícímu na této páteřní síti vědom i spojnice mezi routery **B** a **C** (v tabulce uvozeny písmenkem „O“), přičemž k této spojnici se lze dostat výše zmíněnými cestami.

V tomto prvním případě je uvedena i legenda ke čtení směrovací tabulky (význam uvozujících písmen), v dalším textu už však bude úsporně vynechána.

Bod 2) a 3) zkontrolujeme výpisem ze směrovacích tabulek jednotlivých VRF instancí, pro jednoduchost si porovnejme výstup pro VPN_A na směrovači **A**, přičemž zbývající tabulky obsahují analogické údaje:

```
A#show ip route vrf VPN_A

Gateway of last resort is 172.16.0.1 to network 0.0.0.0

     10.0.0.0/8 is variably subnetted, 9 subnets, 3 masks
O       10.1.10.0/24 [110/2] via 10.1.210.2, 00:01:50, FastEthernet0/0/1.210
                     [110/2] via 10.1.110.2, 00:03:30, FastEthernet0/0/0.110
B       10.2.10.0/24 [200/2] via 192.168.2.2, 19:46:48
B       10.3.10.0/24 [200/0] via 192.168.3.3, 2d04h
B       10.2.1.1/32 [200/0] via 192.168.2.2, 19:52:03
B       10.3.1.1/32 [200/0] via 192.168.3.3, 2d04h
C       10.1.1.1/32 is directly connected, Loopback1
C       10.1.110.0/30 is directly connected, FastEthernet0/0/0.110
B       10.2.110.0/24 [200/0] via 192.168.2.2, 19:52:03
C       10.1.210.0/30 is directly connected, FastEthernet0/0/1.210
B*   0.0.0.0/0 [20/0] via 172.16.0.1 (VPN_OVERLAY), 2d04h
```

Router **A** je přímo připojen k jednomu logickému rozhraní (Loopback1) a dvěma rozhraním fyzickým (spojnice na **SW_A1** a **SW_A2**) a právě těmi vede cesta ze směrovače **A** k interní síti klientů ve VPN_A, což se router dozvěděl z IGP OSPF směrovacího procesu. O zbývajících sítích v jiných lokalitách ví **A** díky redistribuci z BGP (cesty označeny „B“).

V hranatých závorkách u jednotlivých směrovacích informací k daným sítím jsou uvedena dvě čísla oddělena lomítkem. Na prvním místě AD (= *administrative distance*), která určuje míru, s jakou router důvěřuje dané cestě, přičemž při směrování preferuje v případě více cest ke stejné síti ty s menší AD. Hodnota AD je determinována platformou a programovým vybavením aktivního síťového prvku, v některých případech se dá i změnit. Na druhém místě je metrika, která obecně určuje kvalitativní charakteristiky cesty (např. počet hopů, cena, šířka pásma, spolehlivost). Metrika je buď nastavitelná, nebo podléhá způsobu zpracování, které jsou určeny tím, jaký směrovací protokol s ní nakládá.

Splnění Bodu 4), tedy dostupnost pro klienty do Internetu, je potvrzeno přítomností cesty s označením „*“ v jednotlivých VRF instančních směrovacích tabulkách. Tato cesta nám vždy říká, jaká je adresa výchozí brány, v případě, že router podle informací, které má, neví, kam by měl směrovat. Hlavní bránou je router **GW**, který má ve své konfiguraci statické směrování z vnitřní sítě a do ní zpět z venku:

```
ip route 0.0.0.0 0.0.0.0 85.239.80.113
ip route 10.0.0.0 255.0.0.0 FastEthernet0/1
```

Protože je směrovač **A** jediný, kdo má spojení na **GW**, je zde zaplé statické směrování pomocí následujícího příkazu. Ve výpisu směrovací tabulky na další stránce je pak tato cesta předcházena písmenkem „S*“:

```
ip route vrf VPN_OVERLAY 0.0.0.0 0.0.0.0 172.16.0.1
```

Od **A** je pak hierarchicky distribuována informace o výchozí bráně do zbývajících koutů topologie. Což dokládá např. část výpisu směrovací tabulky u směrovače **B**:

```
B#show ip route vrf VPN_A
Gateway of last resort is 192.168.1.1 to network 0.0.0.0
B*    0.0.0.0/0 [200/0] via 192.168.1.1, 2d04h
```

K úspěšnému připojených klientů v různých VPN do Internetu je nutnou podmínkou existence zastřešující VRF, která má komplexní směrovací informace o všech příslušejících VRF instancích. V našem případě této funkce na routeru **A** zastává VPN_OVERLAY. Ovšem je nutné ošetřit přístup k této VPN_OVERLAY tak, aby se s její pomocí nedařilo klientům ležícím v různých VPN komunikovat mezi sebou díky statickému směrování na **GW**. Toho bylo dosaženo aplikací rozšířeného ACL na vstup rozhraní FastEthernet0/1 na routeru GW, který zamezuje, aby se do něj dostaly pakety, jež mají zdrojovou i cílovou adresu z rozsahu, který je používán v naší topologii.

```
interface FastEthernet0/1
 ip access-group RESTRIKCE in
ip access-list extended RESTRIKCE
 deny   ip 10.0.0.0 0.255.255.255 10.0.0.0 0.255.255.255
 permit ip any any
```

Na závěr uveďme směrovací tabulku pro VPN_OVERLAY na routeru **A**:

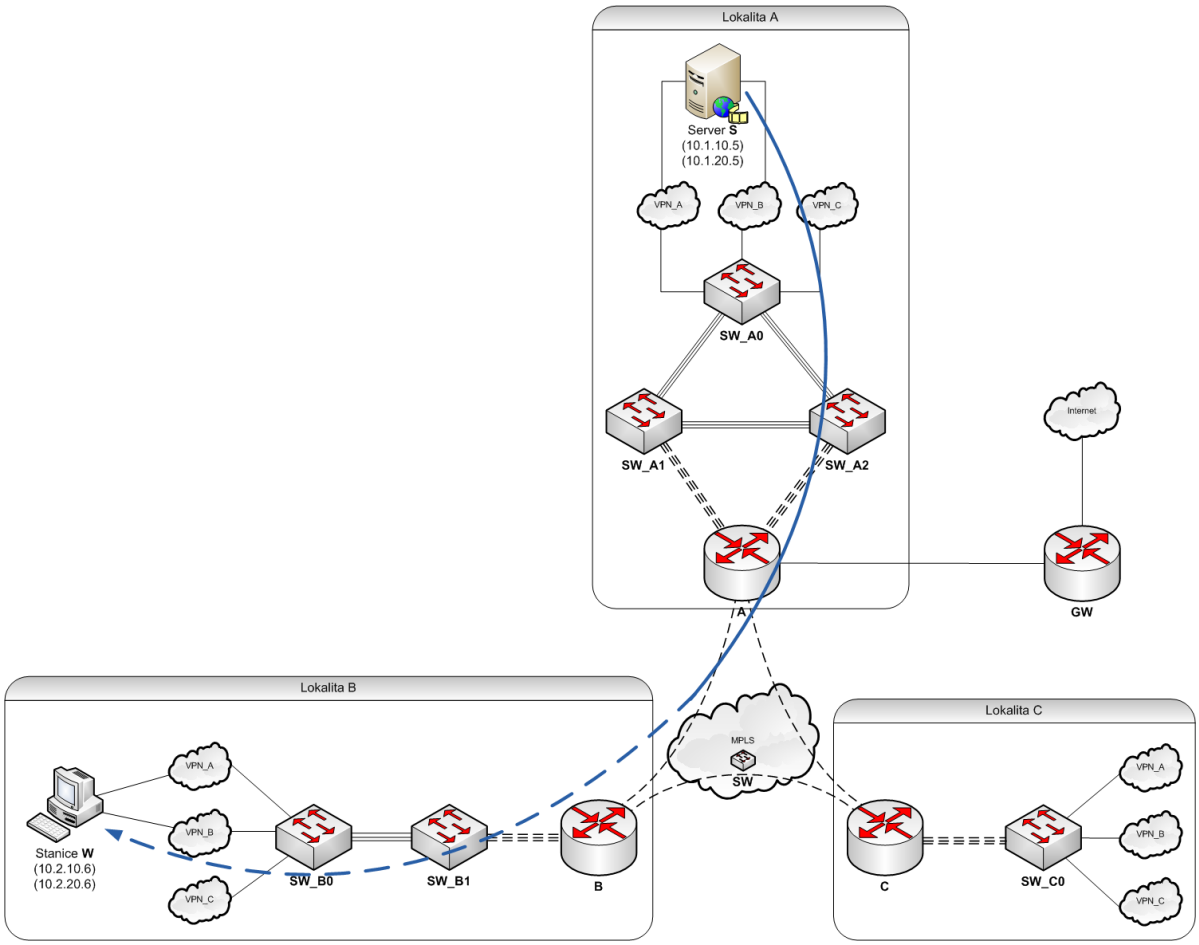
```
A#show ip route vrf VPN_OVERLAY

     Gateway of last resort is 172.16.0.1 to network 0.0.0.0

     172.16.0.0/30 is subnetted, 1 subnets
C       172.16.0.0 is directly connected, GigabitEthernet0/1
     10.0.0.0/8 is variably subnetted, 24 subnets, 3 masks
B       10.1.10.0/24
           [20/2] via 10.1.110.2 (VPN_A), 00:09:34, FastEthernet0/0/0.110
B       10.2.10.0/24 [200/2] via 192.168.2.2, 00:01:34
B       10.3.10.0/24 [200/0] via 192.168.3.3, 2d04h
B       10.2.1.1/32 [200/0] via 192.168.2.2, 19:58:09
B       10.1.2.1/32 is directly connected, 06:14:51, Loopback2
B       10.3.1.1/32 [200/0] via 192.168.3.3, 2d04h
B       10.3.2.1/32 [200/0] via 192.168.3.3, 06:07:36
B       10.2.2.1/32 [200/0] via 192.168.2.2, 06:13:06
B       10.1.1.1/32 is directly connected, 20:02:55, Loopback1
B       10.1.30.0/24
           [20/2] via 10.1.130.2 (VPN_C), 00:09:20, FastEthernet0/0/0.130
B       10.2.30.0/24 [200/2] via 192.168.2.2, 00:00:35
B       10.3.30.0/24 [200/0] via 192.168.3.3, 2d04h
B       10.2.20.0/24 [200/2] via 192.168.2.2, 00:00:50
B       10.3.20.0/24 [200/0] via 192.168.3.3, 2d04h
B       10.1.20.0/24
           [20/2] via 10.1.120.2 (VPN_B), 00:09:20, FastEthernet0/0/0.120
B       10.1.110.0/30 is directly connected, 00:12:36, FastEthernet0/0/0.110
B       10.2.110.0/30 [200/0] via 192.168.2.2, 00:03:05
B       10.2.120.0/30 [200/0] via 192.168.2.2, 00:02:05
B       10.1.120.0/30 is directly connected, 00:12:21, FastEthernet0/0/0.120
B       10.1.130.0/30 is directly connected, 00:12:05, FastEthernet0/0/0.130
B       10.2.130.0/30 [200/0] via 192.168.2.2, 00:02:05
B       10.1.220.0/30 is directly connected, 00:09:20, FastEthernet0/0/1.220
B       10.1.210.0/30 is directly connected, 00:09:35, FastEthernet0/0/1.210
B       10.1.230.0/30 is directly connected, 00:09:20, FastEthernet0/0/1.230
S*   0.0.0.0/0 [1/0] via 172.16.0.1
```

Úspěšný NAT překlad adres na **GW**, lze ověřit příslušným výpisem:

```
GW#show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
icmp 85.239.80.114:0   10.1.110.2:0       147.229.9.23:0     147.229.9.23:0
udp 85.239.80.114:55241 10.2.10.2:55241   78.45.35.206:69    78.45.35.206:69
udp 85.239.80.114:55241 10.2.10.2:55241   78.45.35.206:59460 78.45.35.206:59460
udp 85.239.80.114:64162 10.3.10.4:64162   78.45.35.206:69    78.45.35.206:69
udp 85.239.80.114:64162 10.3.10.4:64162   78.45.35.206:59459 78.45.35.206:59459
```

5.2 Testovací scénář

Pro multicastové testy uváděné v této práci byl zvolen jednotný scénář, kdy se v lokalitě A nachází vždy zdroj multicastových dat (server **S**), v lokalitě B se nachází odběratel těchto dat (stanice **W**) a v lokalitě C se žádný odběratel nenachází, tak jak ilustruje Obrázek 26:



Obrázek 26 (testovací scénář)

Pro práci s multicastovými daty na serveru a stanici byl zvolen program VLCPlayer v0.9.9 [26], který umožňuje snadno streamovat a přijímat multimediální obsah pomocí protokolu RTP [27] přes multicasting. V rámci multicastingu v jednotlivých lokalitách byl pro přehlednost rozšířen adresní plán o následující adresy:

Adresa/Maska	VRF	Popis
239.1.0.0/16		Multicastová data se zdrojem v lokalitě A
239.2.0.0/16		Multicastová data se zdrojem v lokalitě B
239.3.0.0/16		Multicastová data se zdrojem v lokalitě C

Tabulka 14 (multicastový adresní plán)

5.3 Test č. 1 – PIM-SDM a manuální RP

Pro test je využijeme výše uvedeného scénář, ve kterém se nachází v lokalitě A ve VPN_A server s adresou 10.1.10.5, který multicastově distribuuje videozáznamy na adresu 239.1.10.5 a v lokalitě B je příjemce, který se přihlásí k odběru dat (s adresou 10.2.10.6).

Popišme si nyní situaci, která nastane při použití PIM-SDM. Když spustíme multicasting a na žádném *mrouteru* (přičemž výrazem mrouter budeme od teď chápat buď přepínač, nebo směrovač se schopností multicastového směrování) není nijak nakonfigurován žádný RP pro libovolnou skupinu, což dokazuje následující výpis:

PIM-SDM bude pracovat v dense módu a podívejme se nyní na multicastovou směrovací tabulku na routeru **A**:

```
A#show ip mroute vrf VPN_A
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group
Outgoing interface flags: H - Hardware switched, A - Assert winner
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode
(*, 239.1.10.5), 00:01:49/stopped, RP 0.0.0.0, flags: D
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    FastEthernet0/0/1.210, Forward/Sparse-Dense, 00:01:49/00:00:00
    FastEthernet0/0/0.110, Forward/Sparse-Dense, 00:01:49/00:00:00
    Tunnel0, Forward/Sparse-Dense, 00:01:49/00:00:00

(10.1.10.5, 239.1.10.5), 00:01:49/00:01:40, flags: T
  Incoming interface: FastEthernet0/0/1.210, RPF nbr 10.1.210.2
  Outgoing interface list:
    FastEthernet0/0/0.110, Forward/Sparse-Dense, 00:01:50/00:00:00
    Tunnel0, Forward/Sparse-Dense, 00:01:50/00:00:00

(*, 224.0.1.40), 00:01:53/00:02:36, RP 0.0.0.0, flags: DCL
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    FastEthernet0/0/1.210, Forward/Sparse-Dense, 00:01:53/00:00:00
    FastEthernet0/0/0.110, Forward/Sparse-Dense, 00:01:53/00:00:00
    Tunnel0, Forward/Sparse-Dense, 00:01:53/00:00:00
```

Začněme od konce, skupina 224.0.1.40 je skupina, pomocí které si PIM routery předávají informace o RP v případě automatizované metody distribuce RP jako je proprietálně Auto-RP [28] od Ciscoa. Vyskytuje se tedy automaticky (pokud není vinou špatné konfigurace zakázána např. pomocí ACL), avšak v rámci našeho testování nemá žádný vliv, proto je uvedena jen v tomto prvním výpisu, přičemž s úsporných důvodů je ve zbývajících výpisech příkazu `show ip mroute` vynechávána, stejně jako legenda k příkazu.

(*, 239.1.10.5) je záznam pro multicastovou skupinu operující v dense módu (označeno příznakem „D“), kde není žádný rendezvous point (RP je nastaven na 0.0.0.0), přičemž následuje soupiska rozhraní, kterými je tento strom rozšiřován dále. Záznam (10.1.10.5, 239.1.10.5) s příznakem „T“ označuje zdrojová distribuční strom, který je SPT ke zdroji multicastingu (v našem případě serveru **S**).

Nyní se stanice **W** ve Vlan10 rozhodne přihlásit k odběru multicastových dat pomocí zprávy *IGMP Membership*, kterou správně detekuje **SW_B1**:

```
SW_B1#show ip igmp vrf VPN_A groups
IGMP Connected Group Membership
Group Address    Interface                Uptime    Expires   Last Reporter   Group
Accounted
239.255.255.250  Vlan10                   00:31:33  00:02:42  10.2.10.6
239.1.10.5       Vlan10                   00:21:12  00:02:40  10.2.10.6
224.0.1.40       Vlan110                  1w5d      00:02:51  10.2.110.2
```

A následuje výpis z multicastové směrovací tabulky:

```
SW_B1#show ip mroute vrf VPN_A
(*, 239.1.10.5), 00:06:49/stopped, RP 0.0.0.0, flags: DC
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Vlan10, Forward/Sparse-Dense, 00:06:49/00:00:00
    Vlan110, Forward/Sparse-Dense, 00:06:49/00:00:00

(10.1.10.5, 239.1.10.5), 00:01:34/00:02:55, flags: T
  Incoming interface: Vlan110, RPF nbr 10.2.110.1
  Outgoing interface list:
    Vlan10, Forward/Sparse-Dense, 00:01:34/00:00:00
```

Protože se v lokalitě B nachází příjemce multicastingu (stanice **W**), tak se k ní nejbližší mrouter **SW_B1** napojí na (*, 239.1.10.5), což indikuje příznak „DC“. A zároveň s tím začne odebírat multicastová data ze zdrojového stromu (10.1.10.5, 239.1.10.5).

Na závěr si ještě uveďme výpis ze směrovací tabulky na lokalitě, kde se nenachází žádný odběratel multicastových dat, a to je mrouter **C**:

```
C#show ip mroute vrf VPN_A
(*, 239.1.10.5), 00:00:25/stopped, RP 0.0.0.0, flags: D
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Tunnel2, Forward/Sparse-Dense, 00:00:25/00:00:00

(10.1.10.5, 239.1.10.5), 00:00:25/00:02:34, flags: PT
  Incoming interface: Tunnel2, RPF nbr 192.168.1.1
  Outgoing interface list: Null
```

Lokalita C díky pravidelným záplavovým výzvám PIM-DM ví o tom, že existuje distribuční strom (*, 293.1.10.5) operující v dense módu. Ale protože se zde nenachází žádný odběratel, tak není připojena k příslušejícímu zdrojovému stromu (10.1.10.5, 239.1.10.5), kterým by proudila samotná data, což je označeno příznakem „P“ (*pruned*).

Nyní zastavíme streamování na serveru **S**, provedeme resetování multicastových směrovacích tabulek na jednotlivých zařízeních příkazem `clear ip mroute vrf VPN_A *` a nakonfigurujeme manuálně statické RP na všech mrouterech v topologii, a to pomocí následující sekvence příkazů:

```
ip pim vrf VPN_A rp-address 10.1.1.1 MCAST-A
ip pim vrf VPN_A rp-address 10.2.1.1 MCAST-B
ip pim vrf VPN_A rp-address 10.3.1.1 MCAST-C
ip access-list standard MCAST-A
 permit 239.1.0.0 0.0.255.255
ip access-list standard MCAST-B
 permit 239.2.0.0 0.0.255.255
ip access-list standard MCAST-C
 permit 239.3.0.0 0.0.255.255
```

Samotné namapování RP na skupinu se provádí příkazem `ip pim rp-address`, jehož prvním vstupním argumentem po určení VRF instance je IP adresa rendezvous pointu a druhým je ACL, kterým máme možnost vložit celý rozsah multicastových skupin, pro které je vkládané zařízení RP. Z praktických i bezpečnostních důvodů byly vytvořeny tři ACL v dohodnutém rozsahu, které jsou pak výše uvedeným příkazem provázány vždy s adresou páteřního routeru v dané lokalitě.

Úspěch při této manuální konfiguraci si lze ověřit následovně:

```
SW_B1#show ip pim vrf VPN_A rp mapping
PIM Group-to-RP Mappings

Acl: MCAST-A, Static
    RP: 10.1.1.1 (?)
Acl: MCAST-B, Static
    RP: 10.2.1.1 (?)
Acl: MCAST-C, Static
    RP: 10.3.1.1 (?)
```

Obnovíme streamování videa na serveru **S**, ovšem ještě nepřihlásíme stanici **W** k odběru. Opět si nejprve ukážeme situaci na routeru v lokalitě, v níž se nachází zdroj, tedy **A**:

```
A#show ip mroute vrf VPN_A
(*, 239.1.10.5), 00:00:04/stopped, RP 10.1.1.1, flags: SP
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list: Null

(10.1.10.5, 239.1.10.5), 00:00:04/00:02:55, flags: P
  Incoming interface: FastEthernet0/0/1.210, RPF nbr 10.1.210.2
  Outgoing interface list: Null
```

Router **A** přijal od **SW_A2**, který detekoval zdroj multicastu, zprávu *PIM Register*, založil si sdílený distribuční strom (*, 239.1.10.5), který pracuje ve sparse módu (označení písmenkem „S“), pro nějž ví, že je i rendezvous pointem. Protože však ještě není nikde na síti přihlášen žádný odběratel, tak tento strom není aktivní a samotný RP se od něj odřeže (dodatečné písmenko „P“), což jen potvrzuje, že není ve výpisu u této cesty uvedeno žádné odchozí ani příchozí rozhraní. Přihlásíme nyní stanici **W** ve Vlan10 k odběru dat, jaká je situace na routeru **SW_B1**:

```
SW_B1#show ip mroute vrf VPN_A
(*, 239.1.10.5), 00:00:44/00:02:24, RP 10.1.1.1, flags: SJC
  Incoming interface: Vlan110, RPF nbr 10.2.110.1
  Outgoing interface list:
    Vlan10, Forward/Sparse-Dense, 00:00:44/00:02:46

(10.1.10.5, 239.1.10.5), 00:00:44/00:02:54, flags: JT
  Incoming interface: Vlan110, RPF nbr 10.2.110.1
  Outgoing interface list:
    Vlan10, Forward/Sparse-Dense, 00:00:44/00:02:46
```

SW_B1 detekoval příjemce multicastu 239.1.10.5 na jednom ze svých rozhraní, a protože pro tuto skupinu zná RP (což je mrouter **A** s adresou 10.1.1.1), tak se k němu připojí pomocí zprávy *PIM Join*. **A** detekuje prvního odběratele, takže se serverem **S** spojí vlastním zdrojovým stromem a začne distribuovat multicastová data sdíleným distribučním stromem (*, 239.1.10.5), ve výpisu níže označeno zmizel příznak „P“. **B** se tak může úspěšně připojit do tohoto sdíleného stromu (ve výpisu na výše označeno příznakem „SJC“). Zdrojový SPT strom s příznakem „JT“ výše mrouter **B** použije v případě, kdy by bylo nutné sebeoptimalizovat spojení se zdrojem multicastových dat, tak jak bylo popsáno v závěru kapitoly 3.4.3 PIM – Sparse Mode.

```
A#show ip mroute vrf VPN_A
(*, 239.1.10.5), 00:04:48/00:02:56, RP 10.1.1.1, flags: S
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Tunnel0, Forward/Sparse-Dense, 00:03:30/00:02:56

(10.1.10.5, 239.1.10.5), 00:04:48/00:03:29, flags: T
  Incoming interface: FastEthernet0/0/1.210, RPF nbr 10.1.210.2
  Outgoing interface list:
    Tunnel0, Forward/Sparse-Dense, 00:03:30/00:02:56
```

Na závěr celého scénáře si ještě uveďme, jak vypadá směrovací tabulka routeru v lokalitě, kde žádný odběratel není, tedy na **C**:

```
C#sh ip mroute vrf VPN_A

(*, 239.1.10.5), 00:01:38/00:02:50, RP 10.1.1.1, flags: SP
  Incoming interface: Tunnel2, RPF nbr 192.168.1.1
  Outgoing interface list: Null
```

Router **C** tedy ví, že existuje v multicastové struktuře sdílený distribuční strom (*, 239.1.10.5) a že cesta k němu vede přes rozraní Tunnel2, ale je od něho odříznut (označeno příznakem „SP“).

5.4 Test č. 2 – PIM-SM a Bootstrap

Jak odhalil Test č. 1, PIM-SDM funguje za každých okolností – ať už je, či není RP přítomen. Co se zdá jako výhoda, může být v případě, že chceme mít síť plně zabezpečenou a odolnou vůči vnějším vlivům problém. Na síti se totiž může vyskytnout nevyžádaný zdroj multicastu, který svým nastavením zahltí všechny prostředky aktivních síťových zařízení. Proto si pro tento test zvolíme jiný mód činnosti PIMu, a to PIM-SM, který bez znalosti RP nemůže správně fungovat.

Ovšem předchozí test ukázal, že ruční nastavování je ve velkých multicastových systémech značně namáhavé a postrádá flexibilitu a škálovatelnost, což je pro PIM-SM, který bez RP nemůže fungovat, překážkou. Nastavíme tedy páteřní routery tak, aby si pomocí Bootstrap mechanismu vyměňovali patřičné informace o RP automaticky, k tomu budou potřeba dva příkazy `ip pim bsr-candidate` a `ip pim rp-candidate`.

Prvním z nich se z routeru stane C-BSR s příslušnými parametry – ilustrováno pro všechny tři směrovače **A**, **B** a **C**:

```
A(config)#ip pim vrf VPN_B bsr-candidate Loopback2 30 192
B(config)#ip pim vrf VPN_B bsr-candidate Loopback2 30
C(config)#ip pim vrf VPN_B bsr-candidate Loopback2 30
```

Prvním argumentem je rozhraní, jehož adresa bude sloužit Bootstrapu jako zdroj inzerování, z praktických důvodů byl tedy zvolen logický Loopback2. Druhým argumentem je délka hashové masky, která se používá v případě, že je pro jednu skupinu více různých RP. Třetím je pak explicitní nastavení priority pro volbu BSR. U priority platí pravidlo, že čím je vyšší, tím má směrovač větší šanci být zvolen BSR. V případě, že více směrovačů sdílí stejnou prioritu, tak při volbě vyhrává ten s větší (binárně vyšší) IP adresou. Dle RFC 5059 [14], by měla být implicitní hodnota priority 64, toho se však zařízení od firmy Cisco nedrží a mají výchozí hodnotu 0. Nastavením priority pro směrovač **A** na 192, bylo dosaženo toho, že pokud nedojde k jeho výpadku, bude v rámci naší topologie vždy zvolen jako BSR.

Druhým příkazem nastavíme, že pro jednotlivé páteřní routery, aby se staly C-RP pro multicastové skupiny využívající zdrojové adresy příslušející jejich lokalitě. K určení rozsahu inzerovaných skupin znovupoužijeme přístupových pravidel, která jsme si vytvořili v Testu č. 1.

```
A(config)#ip pim vrf VPN_B rp-candidate Loopback2 group-list MCAST-A
B(config)#ip pim vrf VPN_B rp-candidate Loopback2 group-list MCAST-B
C(config)#ip pim vrf VPN_B rp-candidate Loopback2 group-list MCAST-C
```

Elekci BSR routeru a tedy úspěšně začlenění páteřních routerů do Bootstrap mechanismu, stejně jako inzerování RP si můžeme ověřit z dvou výpisů např. na mrouteru **C**:

```
C#show ip pim vrf VPN_B bsr-router
PIMv2 Bootstrap information
  BSR address: 10.1.2.1 (?)
  Uptime:      01:42:53, BSR Priority: 192, Hash mask length: 30
  Expires:     00:01:51
This system is a candidate BSR
  Candidate BSR address: 10.3.2.1, priority: 0, hash mask length: 30
  Candidate RP: 10.3.2.1(Loopback2)
    Holdtime 150 seconds
    Advertisement interval 60 seconds
    Next advertisement in 00:00:21
    Group acl: MCAST-C

C#show ip pim vrf VPN_B rp mapping
PIM Group-to-RP Mappings
This system is a candidate RP (v2)
Group(s) 239.1.0.0/16
  RP 10.1.2.1 (?), v2
    Info source: 10.1.2.1 (?), via bootstrap, priority 0, holdtime 150
         Uptime: 14:07:26, expires: 00:01:49
Group(s) 239.2.0.0/16
  RP 10.2.2.1 (?), v2
    Info source: 10.1.2.1 (?), via bootstrap, priority 0, holdtime 150
         Uptime: 14:06:44, expires: 00:01:50
Group(s) 239.3.0.0/16
  RP 10.3.2.1 (?), v2
    Info source: 10.1.2.1 (?), via bootstrap, priority 0, holdtime 150
         Uptime: 14:06:57, expires: 00:01:50
```

První příkaz nám říká adresu BSR v doméně a že **C** je C-BSR. Dále je **C** také C-RP pro skupinu adres definovaných pomocí listu přístupových pravidel MCAST-C. Výstup druhého příkazu je společný všem PIM routerům v doméně a informuje o RP pro jednotlivé multicastové skupiny, stejně tak i o tom, kdo je zdrojem těchto informací (BSR směrovač **A**).

Pro tento test využijeme stejného scénáře, jen s tou obměnou, že nyní budeme pracovat ve VPN_B. Spustíme-li videostreamující server s adresou 10.1.20.5 v lokalitě A a začneme-li odebírat data, která vysílá na adresu 239.1.20.5, stanicí připojenou v lokalitě B s adresou 10.2.20.6, můžeme pozorovat následující změny v multicastovém směrování pro VPN_B. Nejprve výpis z tabulky na směrovači **A**, který leží u zdroje multicastových dat:

```
A#show ip mroute vrf VPN_B
(*, 239.1.20.5), 00:41:28/00:03:15, RP 10.1.2.1, flags: S
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Tunnel1, Forward/Sparse-Dense, 00:30:45/00:03:15
(10.1.20.5, 239.1.20.5), 00:35:02/00:03:26, flags: T
  Incoming interface: FastEthernet0/0/1.220, RPF nbr 10.1.220.2
  Outgoing interface list:
    Tunnel1, Forward/Sparse-Dense, 00:30:45/00:03:16
```

Popišme nyní zevrubně, co se stalo. V lokalitě A byl detekován nový zdroj multicastingu, a to server **S**. Nejbližší mrouter, který zaregistroval tento zdroj (v našem případě **SW_A1**), ví díky Bootstrap mechanismu, že RP pro multicastovou skupinu 239.1.20.5 je router **A**. Přepošle tedy na něj *PIM Register* zprávu. **A** vytvoří sdílený strom (*, 239.1.20.5) pro daný multicast (v uvedené směrovací tabulce označen příznakem „S“) a zároveň sestaví vlastní zdrojový strom (10.1.20.5, 239.1.20.5) k multicastovému zdroji (označen příznakem „T“).

A nyní se podívejme na multicastovou směrovací tabulku na směrovači **B**:

```
SW_B1#sh ip mroute vrf VPN_B
(*, 239.1.20.5), 00:00:57/00:02:09, RP 10.1.2.1, flags: SJC
  Incoming interface: Vlan120, RPF nbr 10.2.120.1
  Outgoing interface list:
    Vlan20, Forward/Sparse, 00:00:57/00:02:52

(10.1.20.5, 239.1.20.5), 00:00:57/00:02:59, flags: JT
  Incoming interface: Vlan120, RPF nbr 10.2.120.1
  Outgoing interface list:
    Vlan20, Forward/Sparse, 00:00:57/00:02:52
```

V lokalitě B byl detekován odběratel multicastu ze skupiny 239.1.20.5, a to stanice **W**, která vyslala *IGMP Membership* k nejbližšímu routeru (**SW_B1**). SW_B1 díky Bootstrapu zjistí, kdo je RP pro danou skupinu v doméně a vyšle k němu zprávu *PIM Join*, čímž se napojí na sdílený strom (*, 239.1.20.5), což je označeno příznakem „SJC“.

A stejně jako minule si na závěr uvedeme výpis z mrouteru **C**, který se nachází v lokalitě bez odběratele:

```
C#show ip mroute vrf VPN_B
(*, 239.1.20.5), 00:04:57/00:02:55, RP 10.1.2.1, flags: SP
  Incoming interface: Tunnel1, RPF nbr 192.168.1.1
  Outgoing interface list: Null
```

Z tabulky vyplývá, že **C** ví o multicastovém sdíleném distribučním stromu (*, 239.1.20.5) pracujícím ve sparse módu, zná jeho RP (10.1.2.1), ale díky absenci odběratele je od něj odříznut.

Bootstrapping je tedy účinný způsob, jak zvětšit škálovatelnost celé topologie zbavením se závislosti na ručním zanášení informace o rozmístění RP, je však nutné přihlédnout i k bezpečnostním aspektům. V hypotetickém scénáři s útočníkem, který má fyzický přístup k síti, může dojít k situaci, kdy by Boostrap mechanismu zneužil ve svůj nekalý prospěch, a to připojením (či emulací) vlastního C-BSR routeru, který by měl takové parametry, že by se stal BSR a začal modifikovat informace o RP pro danou doménu. Standard zmiňuje tyto potencionální hrozby, avšak nechává jejich zabezpečení na návrhářích a správcích sítě. U Cisco zařízení existují dva příkazy, které se snaží tyto problémy řešit na konfigurační úrovni.

První z nich je příkaz `ip pim bsr-border`, který slouží k ochraně elekčního mechanismu Bootstrappingu a zadává se v konfiguraci interfaceu. Jeho nasazením dojde k zastavení šíření a příjmu zpráv *PIM Bootstrap* skrz rozhraní. V rámci naší topologie tak zamezíme případným útočníkům s přístupem do klientské části, aby nějakým způsobem ovlivnili proces volby BSR. Uveďme si příklad pro mrouter **SW_B1**, zbývající mroutery SW_A1 a SW_A2 se nakonfigurují analogicky:

```
SW_B1(config)#interface Vlan 20
SW_B1(config-subif)#ip pim bsr-border
```

Výsledek si můžeme ihned ověřit a to kontrolním výpisem `show ip pim interface`, kde je hraniční rozraní označeno hvězdičkou:

```
SW_B1#show ip pim vrf VPN_B interface

Address          Interface                Ver/   Nbr    Query  DR     DR
                                          Mode   Count  Intvl  Prior
10.2.120.2       Vlan120                  v2/S   1      30     1      10.2.120.2
10.2.20.2        Vlan20                   v2/S * 0      30     1      10.2.20.2
```

Druhým příkazem je `ip pim accept-rp`, který umožňuje pro jednotlivé PIM routery filtrovat inzerovaný RP pro zadaný rozsah skupin. Případný útočník tak ztratí možnost přesměrovávat multicastový provoz. V rámci naší topologie to znamená na všech mrouterech zadat následující sekvenci příkazů (ve které využijeme v Testu č. 1 definovaných ACL), pomocí níž zamezíme, že k multicastovým skupinám z lokality je RP vždy jejich páteřní router:

```
ip pim vrf VPN_B accept-rp 10.1.2.1 MCAST-A
ip pim vrf VPN_B accept-rp 10.2.2.1 MCAST-B
ip pim vrf VPN_B accept-rp 10.3.2.1 MCAST-C
```

5.5 Shrnutí testů

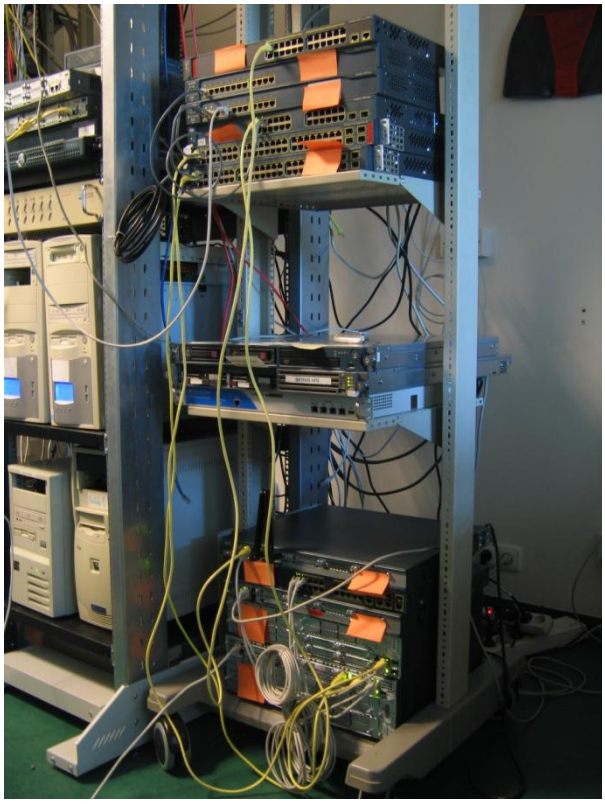
V této kapitole byla požadovaná praktická realizace podrobena několika testům. Nejprve bylo prakticky dokázáno, že výsledek splňuje všechny nemulticastové požadavky – unicastové směrování i vzájemnou konektivitu, stejně jako základní přístupová pravidla.

Byl navržen obvyklý scénář pro vnitropodnikovou multicastovou strukturu v podobě jednoho dedikovaného multicastového serveru, který je zdrojem dat, a klientskými stanicemi, které leží v různých logicky oddělených částech sítě.

Funkčnost multicastové sítě nad takovýmto scénářem pak byla prověřena ve dvou samostatných testech, užívajících různé přístupy k šíření multimediálního obsahu napříč sítí.

V prvním testu byla demonstrována činnost varianty protokolu PIM pracující v sparse-dense módu. V návaznosti na použitou konfiguraci bylo demonstrováno decentralizované šíření v dense módu, které funguje bez větších návrhových zásahů. Snahou o centralizaci v šíření a samozřejmě zvýšení efektivity při distribuci je použití sparse módu, s čímž souvisí nastavování rendezvous pointů v dané doméně, kde právě první test ukázal nejprimitivnější způsob, a to ruční zadávání.

V druhém testu šlo hlavně o zabezpečení a zefektivnění práce multicastingu. Proto byl nakonfigurován čistý sparse mód činnosti PIMu a jako automatizovaný způsob distribuce RP s úspěchem nasazen Bootstrap mechanismus. V rámci zvýšení ochrany sítě pak byly implementovány metodologie (filtrace RP a vymezení PIM domény), které by měly případným útočníkům znemožnit narušení operability multicastu.



Obrázek 27 (fotka výsledné praktické realizace)

6 Multicasting v Internetu

Na závěr je nutné se alespoň zmínit o možnostech, jak oddělené (např. podnikové) sítě vzájemně propojovat, co se týče multicastingu přes Internet.

První a nejlepší možností v tomto směru je spolupráce s poskytovatelem internetového připojení. V případě zájmu klienta o použití multicastingu musí ISP:

- zajistit rozšíření své multicastové domény i o větev, která je spojnicí mezi klientem a poskytovatelem, což defakto znamená povolení šíření nějakého multicastového směrovacího protokolu;
- domluvit se na adresním plánu pro multicast (malé podniky využívají volně dostupných lokálních multicastových adres, zatímco velké subjekty, jako např. finanční instituce mají vlastní globální adresu – viz Tabulka 1).

K multicastovému směrování u ISP lze užít PIMu, ale mnohem častěji se kvůli menší náročnosti na síťová zařízení poskytovatele nasazuje DVMRP [29], ze kterého se pak v příslušném bodě klientské sítě provede redistribuce do interně běžícího PIMu.

Pokud např. z bezpečnostních či jiných důvodů nelze ISP přinutit k nativní podpoře multicastového směrování ISP pro zákazníka, lze využít druhou možnost, a tou je napojení se na internetový multicastový systém MBONE (*Multicast Backbone*). Síť MBONE [30] vznikla v počátku 90. let minulého století v době, kdy ještě neexistovala podpora multicastingu ze strany výrobců aktivních síťových zařízení a byla to právě ona, kdo se významně zasloužil o rozvoj streamování multimediálního obsahu po Internetu. I když rychle ztratila na popularitě, a to díky nativní podpoře multicastingu u ISP a rozmachem VoIP, tak i ve dnech vzniku této práce ji stále tvoří více než 750 serverů, které mezi sebou a svými klienty komunikují čistě multicastově. Pro připojení klientského serveru je třeba zprovoznění tzv. *MBone tunelu*, který vytváří virtuální spojnici napříč sítí, k tomu je však opět nutná jistá součinnost s poskytovatelem Internetu. Pro více informací k tomuto tématu nezbývá, než čtenáře odkázat na zdroje v Internetu [31], v rámci České Republiky poskytuje připojení k MBONE třeba sdružení CESNET [32].

Poslední možností, která však nesouvisí s multicastingem na linkové a síťové vrstvě, je použití multicastu na aplikační úrovni, tedy implementovat jeho hlavní myšlenku do samotných programů tak, že jejich vzájemné propojování bude odstíněno nižším vrstvám ISO/OSI. Takovým příkladem mohou být třeba diskusní kanály přes protokol IRC [33].

7 Závěr

Od doby svého vzniku v podobě doktorandské práce S. Deeringa z roku 1991 [37] prodělal multicasting rapidní rozvoj. Původně akademický projekt brzy prokázal svůj potenciál a ovlivnil způsob směrování v rámci moderní sítě. Vývoj postoupil od málo optimalizovaných směrovacích protokolů jako MOSPF a DVMRP až k sofistikovanému PIM-SM (který je dnes nasazen v majoritní většině multicastových architektur). Existence výše jmenovaných prostředků nastartovala vybudování podpory nativního multicastingu na aktivních síťových zařízeních, které vedlo ke vzniku a rozvoji v současnosti tolik populárních technologií jako VoIP, IPTV či videokonferenčních systémů.

V teoretické části práce byly pečlivě zdokumentovány všechny protokoly, které s multicastem souvisí a které lze v současnosti nalézt nasazené v reálných sítích. Důraz byl kladen především na podrobný popis vycházející z dokumentů standardizujících jejich činnost (RFC), kde byl výklad patřičně doplněn i obrázkovými materiály pro větší přehlednost. Většina uváděných situací pak byla referenčně otestována buď na fyzických aktivních síťových prvcích, nebo alespoň ve virtuálních laboratořích za použití sady nástrojů Dynamips [34], Dynagen [35] a GNS3 [36].

Co se týče vytyčených cílů této diplomové práce, tak díky spolupráci s firmou ANECT a.s. byl jakožto součást interního projektu navržen model architektury, jenž byl posléze i prakticky realizován a sestaven. V takto vytvořené netriviální síti pak byly výše uvedené teoretické poznatky skloubeny dohromady v jeden obrovský celek, jehož chování bylo pečlivě pozorováno a zdokumentováno. V této části práce byl kladen důraz na vytvoření podmínek, které jsou si blízké spíš s komerčním než pro akademické účely vykonstruovaným prostředím činnosti počítačové sítě.

Přínosem by měl být podrobný postup pro konfigurace multicastingu v multiVRF prostředí, návrhové vzory uvedené pro jeho zabezpečení provozu a případné rady jak propojit takovouto síť s multicastovou částí Internetu. Byly vytvořeny i vhodné testovací scénáře pro ozkoušení samotné činnosti multicastového provozu, jejichž postup a zhodnocování může případný čitatel využít při analýze vlastní multicastové struktury.

Téma multicastingu v Internetu je velmi rozsáhlé a převyšuje obsahové a prostorové možnosti této diplomové práce. Možné pokračování a navázání lze vidět např. v detailním prozkoumání dnes stále oblíbenějšího SSM (*source specific multicast*) či Bi-dir (*bidirectional*) multicastové tvorby sdílených distribučních stromů či samostatné a dosud pořádně ne zcela dořešené kapitoly multicastingu v IPv6, kde je absence IGMP snoopingu pro tuto verzi IP protokolu jednou z velkých překážek v současné etapě migrace celosvětové sítě Internet z IPv4 na IPv6. A právě v tomto směru by autor chtěl navázat na výsledky svého výzkumu v případném pokračujícím doktorandském studiu.

8 Bibliografie

[1] **Wikipedia.** OSI model. [Online] [Citace: 26. prosinec 2008.] http://en.wikipedia.org/wiki/Iso_osi.

[2] **Stehura, Igor.** *Návrh testov komunikácie so skupinovým adresovaním v IP.* Brno: FIT VUT, 2008.

[3] **Meyer.** RFC 2365 - Administratively Scoped IP Multicast. [Online] Červen 1998. [Citace: 25. prosinec 2008.] http://www.isi.edu/in-notes/rfc2365.txt.

[4] **Plummer, David C.** RFC 826 - An Ethernet Address Resolution Protocol. [Online] Listopad 1982. [Citace: 25. prosinec 2008.] http://www.networksorcery.com/enp/rfc/rfc826.txt.

[5] **Narten T., Nordmark E., Simpson W.** *RFC 2461 - Neighbor Discovery for IP Version 6.* [Online] Prosinec 1998. [Citace: 25. prosinec 2008.] http://www.ietf.org/rfc/rfc2461.txt.

[6] **Cain, B., a další.** RFC 3376 - Internet Group Management Protocol, Version 3. [Online] Říjen 2002. [Citace: 26. prosinec 2008.] http://tools.ietf.org/html/rfc3376.

[7] **Holbrook, H., Cain, B. a Haberman, B.** RFC 4604 - Using IGMPv3 and MLDPv2 for Source-Specific Multicast. [Online] 2006 Srpen. [Citace: 26. prosinec 2008.] http://tools.ietf.org/html/rfc4604.

[8] **R. Vida, Ed. a L. Costa, Ed.** RFC 3810 - Multicast Listener Discovery Version 2 (MLDv2) for IPv6. [Online] Červen 2004. [Citace: 26. prosinec 2008.] http://tools.ietf.org/html/rfc3810.

[9] **Christensen, M., Kimball, K. a Solensky, F.** RFC 4541 - Considerations for IGMP and MLD Snooping Switches. [Online] Květen 2006. [Citace: 26. prosinec 2008.] http://tools.ietf.org/html/rfc4541.

[10] **Cisco Systems.** Multicast in a Campus Network: CGMP and IGMP Snooping. [Online] [Citace: 26. prosinec 2008.] http://www.cisco.com/en/US/products/hw/switches/ps708/products_tech_note09186a00800b0871.shtml#igmp_snooping.

[11] **Fenner, B., a další.** RFC 4601 - Protocol Independent Multicast - Sparse Mode (PIM-SM) : Revised. [Online] Srpen 2006. [Citace: 29. prosinec 2008.] http://tools.ietf.org/html/rfc4601.

[12] **Adams, A., Nicholas, J. a Siadak, W.** RFC 3973 - Protocol Independent Multicast - Dense Mode (PIM-DM) : Revised. [Online] Leden 2005. [Citace: 29. prosinec 2008.] http://tools.ietf.org/html/rfc3973.

[13] **Cisco Systems.** How to Implement a Filtering Policy for Rendezvous Points. [Online] [Citace: 2. květen 2009.] http://www.cisco.com/en/US/tech/tk828/technologies_configuration_example09186a00801cb923.shtml.

[14] **Bhaskar, N., a další.** RFC5059 - Bootstrap Router (BSR) Mechanism for Protocol Independent Multicast (PIM). [Online] Leden 2008. [Citace: 2. květen 2009.] http://tools.ietf.org/html/rfc5059.

[15] **Wu, I. a Eckert, T.** RFC 3488 - Router-port Group Management Protocol (RGMP). [Online] Březen 2003. [Citace: 2. leden 2009.] http://tools.ietf.org/html/rfc3488.

[16] **Odom, Wendell, a další.** *CCIE Routing and Switching Official Exam Certification Guide: Official Exam Certification Guide.* San Jose : Cisco Press, 2006. ISBN 1587201410.
[17] **Fenner, B. a Meyer, D.** RFC 3618 - Multicast Source Discovery Protocol (MSDP). [Online] Říjen 2003. [Citace: 6. leden 2009.] http://www.faqs.org/rfcs/rfc3618.html.
[18] **Bates, T., a další.** RFC 4760 - Multiprotocol Extensions for BGP-4. [Online] Leden 2007. [Citace: 6. leden 2009.] http://tools.ietf.org/html/rfc4760.
[19] **Rekhter, Y., Li, T. a Hares, S.** RFC 4271 - A Border Gateway Protocol 4 (BGP-4). [Online] Leden 2006. [Citace: 06. leden 2009.] http://tools.ietf.org/html/rfc4271.
[20] **Moy, J.** RFC 2328 - OSPF version 2. [Online] Duben 1998. [Citace: 3. leden 2009.] http://tools.ietf.org/html/rfc2328.
[21] —. RFC 1584 - Multicast Extensions to OSPF. [Online] Březen 1994. [Citace: 3. leden 2009.] http://tools.ietf.org/html/rfc1584.
[22] **Thaler, D.** RFC 2715 - Interoperability Rules for Multicast Routing Protocols. [Online] Říjen 1999. [Citace: 3. leden 2009.] http://tools.ietf.org/html/rfc2715.
[23] **Moy, J.** RFC 1585 - MOSPF: Analysis and Experience. [Online] Březen 1994. [Citace: 3. leden 2009.] http://tools.ietf.org/html/rfc1585.
[24] **Wikipedia.** Fletcher's Checksum. [Online] [Citace: 3. leden 2009.] http://en.wikipedia.org/wiki/Fletcher's_checksum.
[25] **Li, T., a další.** RFC 2281 - Cisco Hot Standby Router Protocol (HSRP). [Online] Březen 1998. [Citace: 7. leden 2009.] http://tools.ietf.org/html/rfc2281.
[26] **VideoLAN.** VLC Media Player. [Online] [Citace: 5. květen 2009.] http://www.videolan.org/vlc/.
[27] **Wikipedia.** Real-time Transport Protocol. [Online] [Citace: 5. květen 2009.] http://en.wikipedia.org/wiki/Real-time_Transport_Protocol.
[28] **Cisco Systems.** Multicast Quick-Start Configuration Guide. [Online] [Citace: 5. květen 2009.] http://www.cisco.com/en/US/tech/tk828/technologies_tech_note09186a0080094821.shtml.
[29] **Deering, S., Waitzman, D. a Partridge, C.** Distance Vector Multicast Routing Protocol. [Online] Listopad 1988. [Citace: 7. květen 2009.] http://tools.ietf.org/html/rfc1075.
[30] **Devereaux-Weber, Dave.** mbone.net. [Online] 26. březen 2006. [Citace: 7. květen 2009.] http://www.mbone.net/.
[31] **RIPE.** MBone Euro-FAQ. [Online] 12. prosinec 1997. [Citace: 5. květen 2009.] http://www.ripe.net/ripe/wg/mbone/eu-faq.html.
[32] **CESNET.** MBone videokonference. [Online] [Citace: 7. květen 2009.] http://www.cesnet.cz/videokonference/mbone/.
[33] **Oikarinen, J. a Reed, D.** RFC 1459 - Internet Relay Chat Protocol. [Online] Květen 1993. [Citace: 19. květen 2009.] http://tools.ietf.org/html/rfc1459.
[34] **Fillot, Christophe.** Dynamips - Cisco 7200 Simulator. [Online] [Citace: 07. leden 2009.] http://www.ipflow.utc.fr/index.php/Cisco_7200_Simulator.

[35] **Anuzelli, Greg.** Dynagen - The Network Configuration Generator for Dynamips. [Online] [Citace: 7. leden 2009.] http://www.dynagen.org/.

[36] **Grossmann, Jeremy a Alt, Xavier.** GNS3 - Graphic Network Simulator. [Online] [Citace: 7. leden 2009.] http://www.gns3.net/.

[37] **Deering, Stephan Edward.** Multicast Routing in a Datagram Internetwork. [Online] Prosinec 1991. [Citace: 19. květen 2009.] http://www.dtic.mil/cgi-bin/GetTRDoc?AD=ADA325909&Location=U2&doc=GetTRDoc.pdf.

[38] **Cisco Systems.** Cisco IOS Release 12.0 Network Protocols Configuration Guide, Part 1 - Configuring OSPF. [Online] [Citace: 2. květen 2009.] http://www.cisco.com/en/US/docs/ios/12_0/np1/configuration/guide/1cospf.html.

[39] **Javvin.** Multicast Extension to OSPF. *Javvin - Network Management and Securuity.* [Online] [Citace: 3. leden 2009.] http://www.javvin.com/protocolMOSPF.html.

[40] **Cisco Systems.** RGMP: Basics and Case Study. [Online] [Citace: 2. leden 2009.] http://www.cisco.com/en/US/products/hw/switches/ps700/products_tech_note09186a008011c11b.shtml#topic1.

[41] **Filip, Ondřej.** Úvod do IP multicastu. *LUPA.* [Online] 10. září 2004. [Citace: 27. prosinec 2008.] http://www.lupa.cz/clanky/uvod-do-ip-multicastu/.

[42] **Hinden R., Deering S.** RFC 2375 - IPv6 Multicast Address Assignments. [Online] Červen 1998. [Citace: 25. prosinec 2008.] http://www.rau.edu.uy/ipv6/rfc2375.txt.

9 Přílohy

9.1 Seznam zkratek

Zkratka	Úplný název
IGMP	Internet Group Management Protocol
IP	Internet Protocol
ARP	Address Resolution Protocol
IANA	Internet Assigned Numbers Authority
MAC	Media Access Control
ICMP	Internet Control Message Protocol
RIP	Routing Information Protocol
EIGRP	Enhanced Interior Gateway Routing Protocol
PIM	Protocol Independent Multicast
PIM-SM	PIM sparse mode
PIM-DM	PIM dense mode
PIM-SDM	PIM sparse-dense mode
CGMP	Cisco Group Management Protocol
RGMP	Router Gateway Management Protocol
MSDP	Multicast Source Discovery Protocol
BGP	Border Gateway Protocol
M-BGP	Multicast BGP
SPF	Shortest Path First
OSPF	Open SPF
MOSPF	Multicast OSPF
LSA	Link-state advertisment
DVMRP	Distance Vector Multicast Routing Protocol
RP	Rendezvous point
C-RP	Candidate RP
BSR	Bootstrap router
C-BSR	Candidate BSR
MPLS	Multiprotocol Label Switching
VPN	Virtual Private Network
PE	Providers Edge
CE	Customers Edge
QoS	Quality of Service
HSRP	Hot Standby Router Protocol
VLAN	Virtual LAN
VRF	VPN Routing and Forwarding
RPF	Reverse Path Forwarding

Zkratka	Úplný název
IGP	Interior Gateway Protocol
EGP	Exterior Gateway Protocol
SPT	Shortest Path Tree
MDT	Multicast Distribution Tree
ACL	Access Control List
NAT	Network Address Translation
AD	Administrative distance
TTL	Time to live
NTP	Network Time Protocol
RTP	Real-time Transport Protocol
VoIP	Voice over IP
ISP	Internet Service Provider
MBONE	Multicast Backbone
IPTV	IP Television
SSM	Source specific multicast
Bi-dir	Biderctional
RFC	Request for comments

Tabulka 15 (seznam všech použitých zkratek)

9.2 Seznam obrázků

9.3 Seznam tabulek

9.4 Finální konfigurace

9.4.1 GW

```
version 12.4
service tcp-keepalives-in
service tcp-keepalives-out
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname GW
!
boot-start-marker
boot-end-marker
!
enable password c
!
aaa new-model
!
aaa session-id common
!
resource policy
!
memory-size iomem 5
ip cef
!
no ip domain lookup
ip domain name plocha.net
ip multicast-routing
!
voice-card 0
!
username vvesely privilege 15 password 0
Jesenik1
!
interface FastEthernet0/0
 ip address 85.239.80.114 255.255.255.240
 ip nat outside
 ip virtual-reassembly
 duplex auto
 speed auto
!
interface FastEthernet0/1
 ip address 172.16.0.1 255.255.255.252
 ip access-group RESTRIKCE in
 ip nat inside
 ip virtual-reassembly
 duplex auto
 speed auto
!
ip route 0.0.0.0 0.0.0.0 85.239.80.113
ip route 10.0.0.0 255.0.0.0
FastEthernet0/1
!
ip http server
no ip http secure-server
ip nat pool OVRLD 85.239.80.114
85.239.80.114 prefix-length 28
ip nat inside source list 99 pool OVRLD
overload
!
ip access-list extended RESTRIKCE
 deny   ip 10.0.0.0 0.255.255.255 10.0.0.0
0.255.255.255
 permit ip any any
!
access-list 1 permit 147.229.196.48
access-list 1 permit 78.45.35.206
access-list 1 permit 85.239.80.33
access-list 99 permit 10.0.0.0
0.255.255.255
!
control-plane
!
line con 0
 logging synchronous
line aux 0
line vty 0 4
 password c
 transport input ssh
line vty 5 15
 password c
 transport input ssh
!
scheduler allocate 20000 1000
ntp clock-period 17179851
ntp master 2
ntp server 195.113.144.201
end
```

9.4.2 A

```
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname A
!
boot-start-marker
boot-end-marker
!
enable password c
!
no aaa new-model
memory-size iomem 5
dot11 syslog
no ip dhcp use vrf connected
ip dhcp excluded-address 10.1.10.1
10.1.10.4
ip dhcp excluded-address 10.1.20.1
10.1.20.4
!
ip dhcp pool POOL-A-VPN_A
   network 10.1.10.0 255.255.255.0
   default-router 10.1.1.1
!
ip dhcp pool POOL-A-VPN_B
   network 10.1.20.0 255.255.255.0
   default-router 10.1.2.1
!
ip cef
!
ip vrf VPN_A
 rd 65000:10
 route-target export 65000:10
 route-target import 65000:10
 route-target import 65000:40
 mdt default 239.0.0.10
!
ip vrf VPN_B
 rd 65000:20
 route-target export 65000:20
 route-target import 65000:20
 route-target import 65000:40
 mdt default 239.0.0.20
!
ip vrf VPN_C
 rd 65000:30
 route-target export 65000:30
 route-target import 65000:30
 route-target import 65000:40
 mdt default 239.0.0.30
!
ip vrf VPN_OVERLAY
 rd 65000:0
 export map DEFAULT
 route-target export 65000:0
 route-target import 65000:10
 route-target import 65000:20
 route-target import 65000:30
!
no ip domain lookup
ip multicast-routing
ip multicast-routing vrf VPN_A
ip multicast-routing vrf VPN_B
ip multicast-routing vrf VPN_C
multilink bundle-name authenticated
!
voice-card 0
 no dspfarm
!
archive
 log config
  hidekeys
!
ip tftp source-interface Loopback1
!
interface Loopback0
 ip address 192.168.1.1 255.255.255.255
 ip pim sparse-dense-mode
!
interface Loopback1
 ip vrf forwarding VPN_A
 ip address 10.1.1.1 255.255.255.255
 ip pim sparse-dense-mode
!
interface Loopback2
 ip vrf forwarding VPN_B
 ip address 10.1.2.1 255.255.255.255
 ip pim sparse-mode
!
interface GigabitEthernet0/0
 no ip address
 duplex auto
 speed auto
 media-type rj45
!
interface GigabitEthernet0/0.12
 description === Spojnice A<->B ===
 encapsulation dot1Q 12
 ip address 192.168.12.1 255.255.255.248
 ip pim sparse-dense-mode
 mpls ip
!
interface GigabitEthernet0/0.13
 description === Spojnice A<->C ===
 encapsulation dot1Q 13
 ip address 192.168.13.1 255.255.255.248
 ip pim sparse-dense-mode
 mpls ip
!
interface GigabitEthernet0/1
 ip vrf forwarding VPN_OVERLAY
 ip address 172.16.0.2 255.255.255.252
 duplex auto
 speed auto
 media-type rj45
!
interface FastEthernet0/0/0
 no ip address
 duplex auto
 speed auto
!
interface FastEthernet0/0/0.110
 description === Subinterface k SW_A1 pro
VPN_A ===
 encapsulation dot1Q 110
 ip vrf forwarding VPN_A
 ip address 10.1.110.1 255.255.255.252
 ip pim sparse-dense-mode
!
interface FastEthernet0/0/0.120
 description === Subinterface k SW_A1 pro
VPN_B ===
 encapsulation dot1Q 120
 ip vrf forwarding VPN_B
 ip address 10.1.120.1 255.255.255.252
 ip pim sparse-mode
```

```
!
interface FastEthernet0/0/0.130
 description === Subinterface k SW_A1 pro
VPN_C ===
 encapsulation dot1Q 130
 ip vrf forwarding VPN_C
 ip address 10.1.130.1 255.255.255.252
 ip pim sparse-dense-mode
!
interface FastEthernet0/0/1
 no ip address
 duplex auto
 speed auto
!
interface FastEthernet0/0/1.210
 description === Subinterface k SW_A2 pro
VPN_A ===
 encapsulation dot1Q 210
 ip vrf forwarding VPN_A
 ip address 10.1.210.1 255.255.255.252
 ip pim sparse-dense-mode
!
interface FastEthernet0/0/1.220
 description === Subinterface k SW_A2 pro
VPN_B ===
 encapsulation dot1Q 220
 ip vrf forwarding VPN_B
 ip address 10.1.220.1 255.255.255.252
 ip pim sparse-mode
!
interface FastEthernet0/0/1.230
 description === Subinterface k SW_A2 pro
VPN_C ===
 encapsulation dot1Q 230
 ip vrf forwarding VPN_C
 ip address 10.1.230.1 255.255.255.252
 ip pim sparse-dense-mode
!
router ospf 10 vrf VPN_A
 log-adjacency-changes
 capability vrf-lite
 redistribute bgp 65000 subnets
 network 10.1.0.0 0.0.255.255 area 0
 default-information originate
!
router ospf 20 vrf VPN_B
 log-adjacency-changes
 capability vrf-lite
 redistribute bgp 65000 subnets
 network 10.1.0.0 0.0.255.255 area 0
 default-information originate
!
router ospf 30 vrf VPN_C
 log-adjacency-changes
 capability vrf-lite
 redistribute bgp 65000 subnets
 network 10.1.0.0 0.0.255.255 area 0
 default-information originate
!
router ospf 1
 log-adjacency-changes
 network 192.168.0.0 0.0.255.255 area 0
!
router bgp 65000
 bgp log-neighbor-changes
 neighbor 192.168.2.2 remote-as 65000
 neighbor 192.168.2.2 update-source
Loopback0
 neighbor 192.168.3.3 remote-as 65000
 neighbor 192.168.3.3 update-source
Loopback0
 !
 address-family ipv4
  neighbor 192.168.2.2 activate
  neighbor 192.168.3.3 activate
  no auto-summary
  no synchronization
 exit-address-family
 !
 address-family vpnv4
  neighbor 192.168.2.2 activate
  neighbor 192.168.2.2 send-community both
  neighbor 192.168.3.3 activate
  neighbor 192.168.3.3 send-community both
 exit-address-family
 !
 address-family ipv4 vrf VPN_OVERLAY
  redistribute static
  default-information originate
  no synchronization
 exit-address-family
 !
 address-family ipv4 vrf VPN_C
  redistribute ospf 30 vrf VPN_C route-map
OSPF2BGP
  default-information originate
  no synchronization
 exit-address-family
 !
 address-family ipv4 vrf VPN_B
  redistribute ospf 20 vrf VPN_B route-map
OSPF2BGP
  default-information originate
  no synchronization
 exit-address-family
 !
 address-family ipv4 vrf VPN_A
  redistribute ospf 10 vrf VPN_A route-map
OSPF2BGP
  default-information originate
  no synchronization
 exit-address-family
!
ip forward-protocol nd
ip route vrf VPN_OVERLAY 0.0.0.0 0.0.0.0
172.16.0.1
!
no ip http server
no ip http secure-server
ip pim vrf VPN_A rp-address 10.1.1.1
MCAST-A
ip pim vrf VPN_A rp-address 10.2.1.1
MCAST-B
ip pim vrf VPN_A rp-address 10.3.1.1
MCAST-C
ip pim vrf VPN_B accept-rp 10.1.2.1 MCAST-
A
ip pim vrf VPN_B accept-rp 10.2.2.1 MCAST-
B
ip pim vrf VPN_B accept-rp 10.3.2.1 MCAST-
C
ip pim vrf VPN_B bsr-candidate Loopback2
30 192
ip pim vrf VPN_B rp-candidate Loopback2
group-list MCAST-A
!
ip access-list standard DF-GW
 permit 0.0.0.0
ip access-list standard MCAST-A
 permit 239.1.0.0 0.0.255.255
ip access-list standard MCAST-B
 permit 239.2.0.0 0.0.255.255
ip access-list standard MCAST-C
 permit 239.3.0.0 0.0.255.255
!
access-list 1 permit 10.1.0.0 0.0.255.255
route-map DEFAULT permit 1
 description === Oznacovac na def. GW
spojeni ===
 match ip address DF-GW
 set extcommunity rt  65000:40 additive
!
route-map OSPF2BGP permit 1
 description === Omezeni redistribuce z
OSPF do BGP na lokalne pripojene ===
 match ip address 1
!
control-plane
!
line con 0
 logging synchronous
line aux 0
line vty 0 4
 password c
 logging synchronous
 login
line vty 5 15
 password c
 logging synchronous
 login
!
scheduler allocate 20000 1000
ntp clock-period 17180889
ntp source Loopback1
ntp server vrf VPN_A 172.16.0.1
!
end
```

9.4.3 B

```
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname B
!
boot-start-marker
boot-end-marker
!
enable password c
!
no aaa new-model
memory-size iomem 5
dot11 syslog
no ip dhcp use vrf connected
ip dhcp excluded-address 10.2.10.1
10.2.10.4
ip dhcp excluded-address 10.2.20.1
10.2.20.4
!
ip dhcp pool POOL-B-VPN_A
   network 10.2.10.0 255.255.255.0
   default-router 10.2.1.1
!
ip dhcp pool POOL-B-VPN_B
   network 10.2.20.0 255.255.255.0
   default-router 10.2.2.1
!
ip cef
!
ip vrf VPN_A
 rd 65000:10
 route-target export 65000:10
 route-target import 65000:10
 route-target import 65000:40
 mdt default 239.0.0.10
!
ip vrf VPN_B
 rd 65000:20
 route-target export 65000:20
 route-target import 65000:20
 route-target import 65000:40
 mdt default 239.0.0.20
!
ip vrf VPN_C
 rd 65000:30
 route-target export 65000:30
 route-target import 65000:30
 route-target import 65000:40
 mdt default 239.0.0.30
!
no ip domain lookup
ip multicast-routing
ip multicast-routing vrf VPN_A
ip multicast-routing vrf VPN_B
ip multicast-routing vrf VPN_C
multilink bundle-name authenticated
!
voice-card 0
 no dspfarm
!
archive
 log config
```

```
  hidekeys
!
ip tftp source-interface Loopback1
!
interface Loopback0
 ip address 192.168.2.2 255.255.255.255
 ip pim sparse-dense-mode
!
interface Loopback1
 ip vrf forwarding VPN_A
 ip address 10.2.1.1 255.255.255.255
 ip pim sparse-dense-mode
!
interface Loopback2
 ip vrf forwarding VPN_B
 ip address 10.2.2.1 255.255.255.255
 ip pim sparse-mode
!
interface GigabitEthernet0/0
 no ip address
 duplex auto
 speed auto
 media-type rj45
!
interface GigabitEthernet0/0.12
 description === Spojnice B<->A ===
 encapsulation dot1Q 12
 ip address 192.168.12.2 255.255.255.248
 ip pim sparse-dense-mode
 mpls ip
!
interface GigabitEthernet0/0.23
 description === Spojnice B<->C ===
 encapsulation dot1Q 23
 ip address 192.168.23.2 255.255.255.248
 ip pim sparse-dense-mode
 mpls ip
!
interface GigabitEthernet0/1
 no ip address
 duplex auto
 speed auto
 media-type rj45
!
interface GigabitEthernet0/1.110
 description === Subinterface k SW_B1 pro
VPN_A ===
 encapsulation dot1Q 110
 ip vrf forwarding VPN_A
 ip address 10.2.110.1 255.255.255.252
 ip pim sparse-dense-mode
!
interface GigabitEthernet0/1.120
 description === Subinterface k SW_B1 pro
VPN_B ===
 encapsulation dot1Q 120
 ip vrf forwarding VPN_B
 ip address 10.2.120.1 255.255.255.252
 ip pim sparse-mode
!
interface GigabitEthernet0/1.130
 description === Subinterface k SW_B1 pro
VPN_C ===
 encapsulation dot1Q 130
 ip vrf forwarding VPN_C
 ip address 10.2.130.1 255.255.255.252
 ip pim sparse-dense-mode
!
router ospf 10 vrf VPN_A
 log-adjacency-changes
 capability vrf-lite
 redistribute bgp 65000 subnets
 network 10.2.0.0 0.0.255.255 area 0
 default-information originate
!
router ospf 20 vrf VPN_B
 log-adjacency-changes
 capability vrf-lite
 redistribute bgp 65000 subnets
 network 10.2.0.0 0.0.255.255 area 0
 default-information originate
!
router ospf 30 vrf VPN_C
 log-adjacency-changes
 capability vrf-lite
 redistribute bgp 65000 subnets
 network 10.2.0.0 0.0.255.255 area 0
 default-information originate
!
router ospf 1
 log-adjacency-changes
 network 192.168.0.0 0.0.255.255 area 0
!
router bgp 65000
 bgp log-neighbor-changes
 neighbor 192.168.1.1 remote-as 65000
 neighbor 192.168.1.1 update-source
Loopback0
 neighbor 192.168.3.3 remote-as 65000
 neighbor 192.168.3.3 update-source
Loopback0
 !
 address-family ipv4
  neighbor 192.168.1.1 activate
  neighbor 192.168.3.3 activate
  no auto-summary
  no synchronization
 exit-address-family
 !
 address-family vpnv4
  neighbor 192.168.1.1 activate
  neighbor 192.168.1.1 send-community both
  neighbor 192.168.3.3 activate
  neighbor 192.168.3.3 send-community both
 exit-address-family
 !
 address-family ipv4 vrf VPN_C
  redistribute ospf 30 vrf VPN_C route-map
OSPF2BGP
  no synchronization
 exit-address-family
 !
 address-family ipv4 vrf VPN_B
  redistribute ospf 20 vrf VPN_B route-map
OSPF2BGP
  no synchronization
 exit-address-family
 !
 address-family ipv4 vrf VPN_A
  redistribute ospf 10 vrf VPN_A route-map
OSPF2BGP
  no synchronization
 exit-address-family
!
ip forward-protocol nd
!
no ip http server
no ip http secure-server
ip pim vrf VPN_A rp-address 10.1.1.1
MCAST-A
ip pim vrf VPN_A rp-address 10.2.1.1
MCAST-B
ip pim vrf VPN_A rp-address 10.3.1.1
MCAST-C
ip pim vrf VPN_B accept-rp 10.1.2.1 MCAST-
A
ip pim vrf VPN_B accept-rp 10.2.2.1 MCAST-
B
ip pim vrf VPN_B accept-rp 10.3.2.1 MCAST-
C
ip pim vrf VPN_B bsr-candidate Loopback2
30
ip pim vrf VPN_B rp-candidate Loopback2
group-list MCAST-B
!
ip access-list standard MCAST-A
 permit 239.1.0.0 0.0.255.255
ip access-list standard MCAST-B
 permit 239.2.0.0 0.0.255.255
ip access-list standard MCAST-C
 permit 239.3.0.0 0.0.255.255
!
access-list 1 permit 10.2.0.0 0.0.255.255
route-map OSPF2BGP permit 1
 description === Omezeni redistribuce z
OSPF do BGP na lokalne pripojene ===
 match ip address 1
!
control-plane
!
line con 0
 logging synchronous
line aux 0
line vty 0 4
 password c
 logging synchronous
 login
line vty 5 15
 password c
 logging synchronous
 login
!
scheduler allocate 20000 1000
ntp clock-period 17180730
ntp source Loopback1
ntp server vrf VPN_A 172.16.0.1
!
end
```

9.4.4 C

```
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname C
!
boot-start-marker
boot-end-marker
!
enable password c
!
no aaa new-model
ip cef
!
no ip dhcp use vrf connected
ip dhcp excluded-address 10.3.10.1
10.3.10.4
ip dhcp excluded-address 10.3.20.1
10.3.20.4
!
ip dhcp pool POOL-C-VPN_A
   network 10.3.10.0 255.255.255.0
   default-router 10.3.1.1
!
ip dhcp pool POOL-C-VPN_B
   network 10.3.20.0 255.255.255.0
   default-router 10.3.2.1
!
!
ip vrf VPN_A
 rd 65000:10
 route-target export 65000:10
 route-target import 65000:10
 route-target import 65000:40
 mdt default 239.0.0.10
!
ip vrf VPN_B
 rd 65000:20
 route-target export 65000:20
 route-target import 65000:20
 route-target import 65000:40
 mdt default 239.0.0.20
!
ip vrf VPN_C
 rd 65000:30
 route-target export 65000:30
 route-target import 65000:30
 route-target import 65000:40
 mdt default 239.0.0.30
!
ip tftp source-interface Loopback1
no ip domain lookup
```

```
ip multicast-routing
ip multicast-routing vrf VPN_A
ip multicast-routing vrf VPN_B
ip multicast-routing vrf VPN_C
!
multilink bundle-name authenticated
!
voice-card 0
!
interface Loopback0
 ip address 192.168.3.3 255.255.255.255
 ip pim sparse-dense-mode
!
interface Loopback1
 ip vrf forwarding VPN_A
 ip address 10.3.1.1 255.255.255.255
 ip pim sparse-dense-mode
!
interface Loopback2
 ip vrf forwarding VPN_B
 ip address 10.3.2.1 255.255.255.255
 ip pim sparse-mode
!
interface FastEthernet0/0
 no ip address
 duplex auto
 speed auto
!
interface FastEthernet0/0.13
 description === Spojnice C<->A ===
 encapsulation dot1Q 13
 ip address 192.168.13.3 255.255.255.248
 ip pim sparse-dense-mode
 mpls ip
!
interface FastEthernet0/0.23
 description === Spojnice C<->B ===
 encapsulation dot1Q 23
 ip address 192.168.23.3 255.255.255.248
 ip pim sparse-dense-mode
 mpls ip
!
interface FastEthernet0/1
 no ip address
 duplex auto
 speed auto
!
interface FastEthernet0/1.10
 description === Interni sit ve VPN_A ===
 encapsulation dot1Q 10
 ip vrf forwarding VPN_A
 ip address 10.3.10.1 255.255.255.0
 ip pim sparse-dense-mode
!
interface FastEthernet0/1.20
 description === Interni sit ve VPN_B ===
 encapsulation dot1Q 20
 ip vrf forwarding VPN_B
 ip address 10.3.20.1 255.255.255.0
 ip pim bsr-border
 ip pim sparse-mode
!
interface FastEthernet0/1.30
 description === Interni sit ve VPN_C ===
 encapsulation dot1Q 30
 ip vrf forwarding VPN_C
 ip address 10.3.30.1 255.255.255.0
 ip pim sparse-dense-mode
!
interface Dot11Radio0/3/0
 no ip address
 shutdown
 speed basic-1.0 basic-2.0 basic-5.5 6.0
9.0 basic-11.0 12.0 18.0 24.0 36.0 48.0
54.0
 station-role root
!
interface Dot11Radio0/3/1
 no ip address
 shutdown
 speed basic-6.0 9.0 basic-12.0 18.0
basic-24.0 36.0 48.0 54.0
 station-role root
!
router ospf 1
 log-adjacency-changes
 network 192.168.0.0 0.0.255.255 area 0
!
router bgp 65000
 bgp log-neighbor-changes
 neighbor 192.168.1.1 remote-as 65000
 neighbor 192.168.1.1 update-source
Loopback0
 neighbor 192.168.2.2 remote-as 65000
 neighbor 192.168.2.2 update-source
Loopback0
 !
 address-family ipv4
 neighbor 192.168.1.1 activate
 neighbor 192.168.2.2 activate
 no auto-summary
 no synchronization
 exit-address-family
 !
 address-family vpnv4
 neighbor 192.168.1.1 activate
 neighbor 192.168.1.1 send-community both
 neighbor 192.168.2.2 activate
 neighbor 192.168.2.2 send-community both
 exit-address-family
 !
 address-family ipv4 vrf VPN_C
 redistribute connected route-map OSPF2BGP
 no synchronization
 exit-address-family
 !
 address-family ipv4 vrf VPN_B
 redistribute connected route-map OSPF2BGP
 no synchronization
 exit-address-family
 !
 address-family ipv4 vrf VPN_A
 redistribute connected route-map OSPF2BGP
 no synchronization
 exit-address-family
!
no ip http server
no ip http secure-server
ip pim vrf VPN_A rp-address 10.1.1.1
MCAST-A
ip pim vrf VPN_A rp-address 10.2.1.1
MCAST-B
ip pim vrf VPN_A rp-address 10.3.1.1
MCAST-C
ip pim vrf VPN_B accept-rp 10.1.2.1 MCAST-
A
ip pim vrf VPN_B accept-rp 10.2.2.1 MCAST-
B
ip pim vrf VPN_B accept-rp 10.3.2.1 MCAST-
C
ip pim vrf VPN_B bsr-candidate Loopback2
30
ip pim vrf VPN_B rp-candidate Loopback2
group-list MCAST-C
!
ip access-list standard MCAST-A
 permit 239.1.0.0 0.0.255.255
ip access-list standard MCAST-B
 permit 239.2.0.0 0.0.255.255
ip access-list standard MCAST-C
 permit 239.3.0.0 0.0.255.255
!
access-list 1 permit 10.3.0.0 0.0.255.255
!
route-map OSPF2BGP permit 1
 description === Omezeni redistribuce
konektlych do BGP na lokalne pripojene ===
 match ip address 1
!
control-plane
!
line con 0
 logging synchronous
line aux 0
line vty 0 4
 password c
 logging synchronous
 login
line vty 5 15
 password c
 logging synchronous
 login
!
scheduler allocate 20000 1000
ntp clock-period 17180821
ntp source Loopback1
ntp server vrf VPN_A 172.16.0.1
end
```

9.4.5 SW_A1

```
version 12.2
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname SW_A1
!
boot-start-marker
boot-end-marker
!
enable password c
!
no aaa new-model
system mtu routing 1500
vtp domain cisco
vtp mode transparent
ip subnet-zero
ip routing
no ip domain-lookup
!
ip vrf VPN_A
 rd 65000:10
!
ip vrf VPN_B
 rd 65000:20
!
ip vrf VPN_C
 rd 65000:30
!
ip multicast-routing vrf VPN_A distributed
ip multicast-routing vrf VPN_B distributed
ip multicast-routing vrf VPN_C distributed
!
crypto pki trustpoint TP-self-signed-
1167897472
 enrollment selfsigned
 subject-name cn=IOS-Self-Signed-
Certificate-1167897472
 revocation-check none
 rsakeypair TP-self-signed-1167897472
!
crypto pki certificate chain TP-self-
signed-1167897472
!
spanning-tree mode pvst
spanning-tree extend system-id
!
vlan internal allocation policy ascending
!
vlan 10
 name VPN_A
!
vlan 20
 name VPN_B
!
vlan 30
 name VPN_C
!
vlan 110,120,130
!
ip tftp source-interface Vlan10
!
```

```
interface FastEthernet0/1
 description === Spojnice k A ===
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 110,120,130
 switchport mode trunk
!
interface FastEthernet0/2
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 10,20,30
 switchport mode trunk
!
interface FastEthernet0/3
 description === Spojnice k SW_A0 ===
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 10,20,30
 switchport mode trunk
!
interface FastEthernet0/4
 switchport access vlan 10
!
interface FastEthernet0/5
 switchport access vlan 20
!
interface FastEthernet0/6
 switchport access vlan 30
!
interface Vlan1
 no ip address
!
interface Vlan10
 ip vrf forwarding VPN_A
 ip address 10.1.10.2 255.255.255.0
 ip helper-address 10.1.1.1
 ip pim sparse-dense-mode
 standby 10 ip 10.1.10.1
 standby 10 preempt
!
interface Vlan20
 ip vrf forwarding VPN_B
 ip address 10.1.20.2 255.255.255.0
 ip helper-address 10.1.2.1
 ip pim bsr-border
 ip pim sparse-mode
 standby 20 ip 10.1.20.1
 standby 20 preempt
!
interface Vlan30
 ip vrf forwarding VPN_C
 ip address 10.1.30.2 255.255.255.0
 ip pim sparse-dense-mode
 standby 30 ip 10.1.30.1
 standby 30 preempt
!
interface Vlan110
 description === Subinterface k A pro
VPN_A ===
 ip vrf forwarding VPN_A
 ip address 10.1.110.2 255.255.255.252
 ip pim sparse-dense-mode
!
interface Vlan120
 description === Subinterface k A pro
VPN_B ===
 ip vrf forwarding VPN_B
 ip address 10.1.120.2 255.255.255.252
 ip pim sparse-mode
!
interface Vlan130
 description === Subinterface k A pro
VPN_C ===
 ip vrf forwarding VPN_C
 ip address 10.1.130.2 255.255.255.252
 ip pim sparse-dense-mode
!
router ospf 10 vrf VPN_A
 log-adjacency-changes
 network 10.1.0.0 0.0.255.255 area 0
!
router ospf 20 vrf VPN_B
 log-adjacency-changes
 network 10.1.0.0 0.0.255.255 area 0
!
router ospf 30 vrf VPN_C
 log-adjacency-changes
 network 10.1.0.0 0.0.255.255 area 0
!
ip classless
ip http server
ip http secure-server
ip pim vrf VPN_A rp-address 10.1.1.1
MCAST-A
ip pim vrf VPN_A rp-address 10.2.1.1
MCAST-B
ip pim vrf VPN_A rp-address 10.3.1.1
MCAST-C
!
ip access-list standard MCAST-A
 permit 239.1.0.0 0.0.255.255
ip access-list standard MCAST-B
 permit 239.2.0.0 0.0.255.255
ip access-list standard MCAST-C
 permit 239.3.0.0 0.0.255.255
!
control-plane
!
line con 0
 logging synchronous
line vty 0 4
 password c
 logging synchronous
 login
line vty 5 15
 password c
 logging synchronous
 login
!
ntp clock-period 36032559
ntp source Vlan10
ntp server vrf VPN_A 172.16.0.1
end
```

9.4.6 SW_A2

```
version 12.2
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname SW_A2
!
boot-start-marker
boot-end-marker
!
enable password c
!
no aaa new-model
system mtu routing 1500
vtp domain cisco
vtp mode transparent
ip subnet-zero
ip routing
no ip domain-lookup
!
ip vrf VPN_A
 rd 65000:10
!
ip vrf VPN_B
 rd 65000:20
!
ip vrf VPN_C
 rd 65000:30
!
ip multicast-routing vrf VPN_A distributed
ip multicast-routing vrf VPN_B distributed
ip multicast-routing vrf VPN_C distributed
!
!
crypto pki trustpoint TP-self-signed-
3335526912
 enrollment selfsigned
 subject-name cn=IOS-Self-Signed-
Certificate-3335526912
 revocation-check none
 rsakeypair TP-self-signed-3335526912
!
spanning-tree mode pvst
spanning-tree extend system-id
!
vlan internal allocation policy ascending
!
vlan 10
 name VPN_A
!
vlan 20
 name VPN_B
!
vlan 30
 name VPN_C
!
vlan 210,220,230
!
ip tftp source-interface Vlan10
!
interface FastEthernet0/1
 description === Spojnice k A ===
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 210,220,230
 switchport mode trunk
!
interface FastEthernet0/2
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 10,20,30
 switchport mode trunk
!
interface FastEthernet0/3
 description === Spojnice k SW_A0 ===
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 10,20,30
 switchport mode trunk
!
interface FastEthernet0/4
 switchport access vlan 10
!
interface FastEthernet0/5
 switchport access vlan 20
!
interface FastEthernet0/6
 switchport access vlan 30
!
interface Vlan1
 no ip address
!
interface Vlan10
 ip vrf forwarding VPN_A
 ip address 10.1.10.3 255.255.255.0
 ip helper-address 10.1.1.1
 ip pim sparse-dense-mode
 standby 10 ip 10.1.10.1
!
interface Vlan20
 ip vrf forwarding VPN_B
 ip address 10.1.20.3 255.255.255.0
 ip helper-address 10.1.2.1
 ip pim bsr-border
 ip pim sparse-mode
 standby 20 ip 10.1.20.1
!
interface Vlan30
 ip vrf forwarding VPN_C
 ip address 10.1.30.3 255.255.255.0
 ip pim sparse-dense-mode
 standby 30 ip 10.1.30.1
!
interface Vlan210
 description === Subinterface k A pro
VPN_A ===
 ip vrf forwarding VPN_A
 ip address 10.1.210.2 255.255.255.252
 ip pim sparse-dense-mode
!
interface Vlan220
```

```
 description === Subinterface k A pro
VPN_B ===
 ip vrf forwarding VPN_B
 ip address 10.1.220.2 255.255.255.252
 ip pim sparse-mode
!
interface Vlan230
 description === Subinterface k A pro
VPN_C ===
 ip vrf forwarding VPN_C
 ip address 10.1.230.2 255.255.255.252
 ip pim sparse-dense-mode
!
router ospf 10 vrf VPN_A
 log-adjacency-changes
 network 10.1.0.0 0.0.255.255 area 0
!
router ospf 20 vrf VPN_B
 log-adjacency-changes
 network 10.1.0.0 0.0.255.255 area 0
!
router ospf 30 vrf VPN_C
 log-adjacency-changes
 network 10.1.0.0 0.0.255.255 area 0
!
ip classless
ip http server
ip http secure-server
ip pim vrf VPN_A rp-address 10.1.1.1
MCAST-A
ip pim vrf VPN_A rp-address 10.2.1.1
MCAST-B
ip pim vrf VPN_A rp-address 10.3.1.1
MCAST-C
!
ip access-list standard MCAST-A
 permit 239.1.0.0 0.0.255.255
ip access-list standard MCAST-B
 permit 239.2.0.0 0.0.255.255
ip access-list standard MCAST-C
 permit 239.3.0.0 0.0.255.255
!
control-plane
!
line con 0
 logging synchronous
line vty 0 4
 password c
 logging synchronous
 login
line vty 5 15
 password c
 logging synchronous
 login
!
ntp clock-period 36032803
ntp source Vlan10
ntp server vrf VPN_A 172.16.0.1
end
```

9.4.7 SW_B1

```
version 12.2
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname SW_B1
!
boot-start-marker
boot-end-marker
!
enable password c
!
no aaa new-model
system mtu routing 1500
vtp domain cisco
vtp mode transparent
ip subnet-zero
ip routing
no ip domain-lookup
!
ip vrf VPN_A
 rd 65000:10
!
ip vrf VPN_B
 rd 65000:20
!
ip vrf VPN_C
 rd 65000:30
!
ip multicast-routing vrf VPN_A distributed
ip multicast-routing vrf VPN_B distributed
ip multicast-routing vrf VPN_C distributed
!
!
crypto pki trustpoint TP-self-signed-
239390208
 enrollment selfsigned
 subject-name cn=IOS-Self-Signed-
Certificate-239390208
 revocation-check none
 rsakeypair TP-self-signed-239390208
!
spanning-tree mode pvst
spanning-tree extend system-id
!
vlan internal allocation policy ascending
!
vlan 10
 name VPN_A
!
vlan 20
 name VPN_B
!
vlan 30
 name VPN_C
!
vlan 110,120,130
!
ip tftp source-interface Vlan10
!
interface FastEthernet0/1
 description === Spojnice k B ===
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 110,120,130
 switchport mode trunk
!
interface FastEthernet0/2
!
interface FastEthernet0/3
 description === Spojnice k SW_B0 ===
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 10,20,30
 switchport mode trunk
!
interface FastEthernet0/4
 switchport access vlan 10
!
interface FastEthernet0/5
 switchport access vlan 20
!
interface FastEthernet0/6
 switchport access vlan 30
!
interface Vlan1
 no ip address
!
interface Vlan10
 ip vrf forwarding VPN_A
 ip address 10.2.10.2 255.255.255.0
 ip helper-address 10.2.1.1
 ip pim sparse-dense-mode
 standby 10 ip 10.2.10.1
 standby 10 preempt
!
interface Vlan20
 ip vrf forwarding VPN_B
 ip address 10.2.20.2 255.255.255.0
 ip helper-address 10.2.2.1
 ip pim bsr-border
 ip pim sparse-mode
 standby 20 ip 10.2.20.1
 standby 20 preempt
!
interface Vlan30
 ip vrf forwarding VPN_C
 ip address 10.2.30.2 255.255.255.0
 ip pim sparse-dense-mode
 standby 30 ip 10.2.30.1
 standby 30 preempt
!
interface Vlan110
 description === Subinterface k B pro
VPN_A ===
 ip vrf forwarding VPN_A
 ip address 10.2.110.2 255.255.255.252
 ip pim sparse-dense-mode
!
interface Vlan120
 description === Subinterface k B pro
VPN_B ===
 ip vrf forwarding VPN_B
 ip address 10.2.120.2 255.255.255.252
 ip pim sparse-mode
!
interface Vlan130
 description === Subinterface k B pro
VPN_C ===
 ip vrf forwarding VPN_C
 ip address 10.2.130.2 255.255.255.252
 ip pim sparse-dense-mode
!
router ospf 10 vrf VPN_A
 log-adjacency-changes
 network 10.2.0.0 0.0.255.255 area 0
!
router ospf 20 vrf VPN_B
 log-adjacency-changes
 network 10.2.0.0 0.0.255.255 area 0
!
router ospf 30 vrf VPN_C
 log-adjacency-changes
 network 10.2.0.0 0.0.255.255 area 0
!
ip classless
ip http server
ip http secure-server
ip pim vrf VPN_A rp-address 10.1.1.1
MCAST-A
ip pim vrf VPN_A rp-address 10.2.1.1
MCAST-B
ip pim vrf VPN_A rp-address 10.3.1.1
MCAST-C
!
ip access-list standard MCAST-A
 permit 239.1.0.0 0.0.255.255
ip access-list standard MCAST-B
 permit 239.2.0.0 0.0.255.255
ip access-list standard MCAST-C
 permit 239.3.0.0 0.0.255.255
!
control-plane
!
line con 0
 logging synchronous
line vty 0 4
 password c
 logging synchronous
 login
line vty 5 15
 password c
 logging synchronous
 login
!
ntp clock-period 36032476
ntp source Vlan10
ntp server vrf VPN_A 172.16.0.1
end
```

9.4.8 SW_A0

```
version 12.1
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname SW_A0
!
enable password c
!
ip subnet-zero
!
ip tftp source-interface Vlan10
no ip domain-lookup
ip ssh time-out 120
ip ssh authentication-retries 3
vtp mode transparent
!
spanning-tree mode pvst
no spanning-tree optimize bpdu
transmission
spanning-tree extend system-id
!
vlan 10
 name VPN_A
!
vlan 20
 name VPN_B
!
vlan 30
 name VPN_C
!
interface FastEthernet0/1
 switchport trunk allowed vlan 10,20,30
 switchport mode trunk
!
interface FastEthernet0/2
 switchport trunk allowed vlan 10,20,30
 switchport mode trunk
!
interface FastEthernet0/3
!
interface FastEthernet0/4
 switchport access vlan 10
!
interface FastEthernet0/5
 switchport access vlan 20
!
interface FastEthernet0/6
 switchport access vlan 30
!
interface Vlan1
 no ip address
 no ip route-cache
 shutdown
!
interface Vlan10
 ip address 10.1.10.4 255.255.255.0
 no ip route-cache
!
interface Vlan20
 ip address 10.1.20.4 255.255.255.0
 no ip route-cache
 shutdown
!
interface Vlan30
 ip address 10.1.30.4 255.255.255.0
 no ip route-cache
 shutdown
!
ip http server
!
line con 0
 logging synchronous
line vty 0 4
 password c
 logging synchronous
 login
line vty 5 15
 password c
 logging synchronous
 login
!
ntp source Vlan10
ntp server 172.16.0.1
!
end
```

9.4.9 SW_B0

```
version 12.1
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname SW_B0
!
enable password c
!
ip subnet-zero
!
ip tftp source-interface Vlan10
no ip domain-lookup
ip ssh time-out 120
ip ssh authentication-retries 3
!
spanning-tree mode pvst
no spanning-tree optimize bpdu
transmission
spanning-tree extend system-id
!
interface FastEthernet0/1
 switchport trunk allowed vlan 10,20,30
 switchport mode trunk
!
interface FastEthernet0/2
!
interface FastEthernet0/3
!
interface FastEthernet0/4
 switchport access vlan 10
!
interface FastEthernet0/5
 switchport access vlan 20
!
interface FastEthernet0/6
 switchport access vlan 30
!
interface Vlan1
 no ip address
 no ip route-cache
 shutdown
!
interface Vlan10
 ip address 10.2.10.4 255.255.255.0
 no ip route-cache
!
interface Vlan20
 ip address 10.2.20.4 255.255.255.0
 no ip route-cache
 shutdown
!
interface Vlan30
 ip address 10.2.30.4 255.255.255.0
 no ip route-cache
 shutdown
!
ip http server
!
line con 0
 logging synchronous
line vty 0 4
 password c
 logging synchronous
 login
line vty 5 15
 password c
 logging synchronous
 login
!
ntp source Vlan10
ntp server 172.16.0.1
!
end
```

9.4.10 SW_C0

```
version 12.2
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname SW_C0
!
boot-start-marker
boot-end-marker
!
enable password c
!
no aaa new-model
system mtu routing 1500
ip subnet-zero
!
no ip domain-lookup
!
spanning-tree mode pvst
spanning-tree etherchannel guard misconfig
spanning-tree extend system-id
!
vlan internal allocation policy ascending
!
ip tftp source-interface Vlan10
!
interface FastEthernet0/1
 switchport trunk allowed vlan 10,20,30
 switchport mode trunk
!
interface FastEthernet0/2
!
interface FastEthernet0/3
!
interface FastEthernet0/4
 switchport access vlan 10
!
interface FastEthernet0/5
 switchport access vlan 20
!
interface FastEthernet0/6
 switchport access vlan 30
!
interface Vlan1
 no ip address
 no ip route-cache
 shutdown
!
interface Vlan10
 ip address 10.3.10.4 255.255.255.0
 no ip route-cache
!
interface Vlan20
 ip address 10.3.20.4 255.255.255.0
 no ip route-cache
!
interface Vlan30
 ip address 10.3.30.4 255.255.255.0
 no ip route-cache
!
ip http server
!
control-plane
!
!
line con 0
 logging synchronous
line vty 0 4
 password c
 logging synchronous
 login
line vty 5 15
 password c
```

```
 logging synchronous
 login
!
ntp source Vlan10
ntp server 172.16.0.1
end
```

9.4.11 SW

```
version 12.2
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname SW
!
enable password c
!
no aaa new-model
system mtu routing 1500
ip subnet-zero
!
no ip domain-lookup
!
no file verify auto
spanning-tree mode pvst
 spanning-tree extend system-id
!
vlan internal allocation policy ascending
!
interface GigabitEthernet0/1
 switchport mode trunk
!
interface GigabitEthernet0/2
 switchport mode trunk
!
interface GigabitEthernet0/3
 switchport mode trunk
!
interface GigabitEthernet0/4
!
interface GigabitEthernet0/5
!
interface GigabitEthernet0/6
!
interface GigabitEthernet0/7
!
interface GigabitEthernet0/8
!
 interface GigabitEthernet0/9
!
interface GigabitEthernet0/10
!
interface GigabitEthernet0/11
!
interface GigabitEthernet0/12
!
interface GigabitEthernet0/13
!
interface GigabitEthernet0/14
!
interface GigabitEthernet0/15
!
interface GigabitEthernet0/16
!
interface GigabitEthernet0/17
!
interface GigabitEthernet0/18
!
interface GigabitEthernet0/19
!
interface GigabitEthernet0/20
 !
interface GigabitEthernet0/21
!
interface GigabitEthernet0/22
!
interface GigabitEthernet0/23
!
interface GigabitEthernet0/24
!
interface Vlan1
 no ip address
 no ip route-cache
 shutdown
!
ip http server
!
control-plane
!
line con 0
 logging synchronous
line vty 0 4
 password c
  login
line vty 5 15
 password c
 login
!
end
```