VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

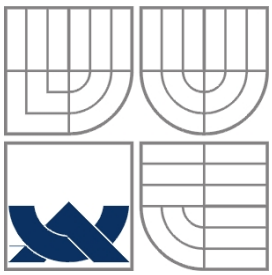
NÁVRH TESTŮ KOMUNIKACE SE SKUPINOVÝM ADRESOVANÍM V IP

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

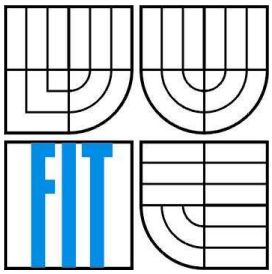
AUTOR PRÁCE
AUTHOR

Bc. IGOR STEHURA

BRNO 2008



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

NÁVRH TESTŮ KOMUNIKACE SE SKUPINOVÝM ADRESOVANÍM V IP

DESIGN OF IP MULTICAST COMMUNICATION TESTS

DIPLOMOVÁ PRÁCE
MASTER‘S THESIS

AUTOR PRÁCE
AUTHOR

Bc. IGOR STEHURA

VEDOUCÍ PRÁCE
SUPERVISOR

prof. Ing. MIROSLAV ŠVÉDA, CSc.

BRNO 2008

Abstrakt

Táto diplomová práca sa zaoberá multicastom. Vysvetlená je adresácia na úrovni druhej a tretej vrstvy ISO/OSI modelu multicastového adresovania. Pre hľadanie optimálnych ciest multicastových paketov sieťou, routre v sieti používajú multicastové smerovacie protokoly, ktorými sa taktiež práca zaoberá. Sú to protokoly DVMRP, protokol PIM vo svojich dvoch režimoch, PIM – Sparse Mode a PIM – Dense Mode. Protokol DVMRP využíva k svojmu správnemu fungovaniu protokol IGMP, ktorý je tiež uvedený. V praktickej časti tejto práce sú uvedené zapojenia, na ktorých boli testy vykonané.

Kľúčové slová

multicast, aplikačný, skupinové adresovanie, multicastová skupina, multicastové protokoly, IGMP, ARP, DVMRP, PIM, hustý režim, riedky režim, IP protokol, SendIGMP, CaptureIGMP, SendMcast, GetMcast.

Abstract

This master's thesis is about mutlicast. There are explained 2nd and 3th layers of the ISO/OSI model multicast addressing. Routers in a network use multicast routing protocols to optimally route multicast packet through the network, this is also in this project. These multicast protocols are DVMRP, protocol PIM in his two modes, PIM – Sparse Mode and PIM – Dense Mode. Protocol DVMRP uses protocol IGMP, which is described as well. At practical section of this master's thesis is presented connections, by which tests was executed.

Keywords

multicast, application, multicast addressing, multicast group, multicast protocols, IGMP, ARP, DVMRP, PIM, Dense Mode, Sparse Mode, IP protocol, Membership Report, Membership Query, SendIGMP, CaptureIGMP, SendMcast, GetMcast.

Citácia

Stehura Igor: Návrh testov komunikácie so skupinovým adresovaním v IP. Brno, 2008, diplomová práca, FIT VUT v Brně.

Návrh testov komunikácie so skupinovým adresovaním v IP

Prehlásenie

Prehlasujem, že som túto diplomovú prácu vypracoval samostatne pod vedením prof. Ing. Miroslava Švédu, CSc.
Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

……
Bc. Igor Stehura
2008-05-05

Poďakovanie

Ďakujem vedúcemu diplomovej práce prof. Ing. Miroslavovi Švédovi, CSc. za cenné rady a pripomienky, ktoré prispeli k vypracovaniu tejto diplomovej práce.

Obsah

1 Úvod

Takmer celá komunikácia v dnešnom Internete je založená na tzv. unicast adresovaní. Objemy prenášaných dát neustále výrazne stúpajú. Určitá časť z nich je ale rovnaká, len sú prenášané viacerým cieľovým staniciam. V tejto situácii, kedy potrebujeme vyslať dáta k viacerým klientom naraz, máme dve možnosti: opakované posielanie alebo broadcast. Opakované posielanie znamená vyslať postupne jednotlivo každej stanici dáta, čo je veľmi neefektívne. Na jednej strane to zaťažuje vysielač dát na druhej strane zahlcuje sieť medzi komunikujúcimi stranami.

Broadcast riešenie predpokladá vyslanie dát všetkým staniciam, bez ohľadu nato, či o tieto dáta majú záujem alebo nie, čo takisto znamená potenciálnu záťaž sieťových zdrojov.

Problémy, ktoré sú spomenuté v predošlých riadkoch je možné vyriešiť pomocou multicastu. Táto technológia bola vyvinutá predovšetkým pre podporu aplikácií prirodzene obsahujúcich komunikáciu jedného zdroja (resp. viacerých zdrojov) dát s veľkým počtom príjemcov.

V rámci prvej časti diplomovej práce sa zaoberám multicastom vo všeobecnosti, základnými pojmami, rozdielom medzi unicastom, broadcastom a multicastom.

Ďalšia kapitola obsahuje základné informácie o skupinovom adresovaní na úrovni linkovej vrstvy a na úrovni IP. Taktiež sa tu zaoberám jednotlivými verziami protokolu IGMP, protokolom ARP a IP.

Štvrtá kapitola informuje o multicastových protokoloch skupiny dense mode a sparse mode, ako aj o distribučných stromoch, tj. zdieľaných a zdrojových.

V piatej kapitole približujem implementáciu multicastu a programy, ktoré som vytvoril za účelom testovania.

Súčasťou šiestej kapitoly sú zapojenia, návrhy testov a popis samotného testovania multicastu.

V prílohe uvádzam konfiguráciu jednotlivých prvkov zapojenia, na ktorom boli vykonávané testy overujúce funkciu vybraných multicastových konfigurácií.

2 Pojmy

Pre lepšiu orientáciu v multicaste uvádzam v tejto kapitole nasledovné pojmy. Popisujem hlavne rozdiely medzi multicastovou komunikáciou a typickou unicastovou komunikáciou, resp. broadcastom.

2.1 Unicast

V počítačových sieťach, unicast, je posielanie informácií, správ (vo všeobecnosti nejakých dát) z jedného zariadenia k jednému cieľovému. Na úrovni IP vrstvy, každý vyslaný paket je posielaný k cieľovému hostu identifikovaného IP adresou cieľa v hlavičke IP paketu. IP routre obsahujú smerovacie tabuľky, ktoré špecifikujú kde, na ktoré rozhranie, majú daný paket preposlať ďalej. Príkladom takejto komunikácie môže byť poslanie mailu jedného užívateľa inému.

2.2 Broadcast

Na rozdiel od unicastu, správy posielané pomocou broadcastu sú poslané každému zariadeniu na sieti. Využitie tejto techniky nadobúda význam vtedy, keď potrebujeme komunikovať so všetkými zariadeniami na sieti, alebo v prípade, kedy komunikujúce zariadenie potrebuje vyslať dáta práve jednému zariadeniu, ale nevie jeho adresu. Príkladom môže byť rádiové vysielanie, ktoré je vyslané každému, bez ohľadu nato, či o to má záujem, alebo nie. Ďalším príkladom môže byť protokol ARP, ktorý popíšem neskôr.

2.3 Multicast

Multicast je kompromis v komunikácii na rozdiel od predchádzajúcich typov komunikácie. Informácie nie sú posielané všetkým zariadeniam ako bolo uvedené v predošlej časti, pretože môžeme zbytočne zaťažovať zariadenia, ktoré o tieto posielané informácie nestoja. Tak isto ale nie sú posielané ani každému zo záujemcov postupne použitím unicastu pre jednotlivé zariadenia. Príkladom multicastu môže byť email poslaný od jedného užívateľa k viacerým.

Ďalším príkladom, v ktorom by multicast mohol zohrať významnú úlohu sú konferencie na Internete. Unicastovým typom komunikácie v konferencii by mohlo dochádzať k veľkému zaťažovaniu. V prípade, že by chcel napríklad riaditeľ firmy komunikovať so skupinou niekoľkých svojich zamestnancov, ktorý by sa nachádzali na druhej strane sveta, musela by byť informácia s každým z nich prenášaná jednotlivo. Naopak, v prípade multicastovej komunikácie, informácia od

riaditeľa by bola poslaná zamestnancom len jeden krát, čo by mohlo výraznou mierou ušetriť výpočetný výkon prostriedkov, ako aj prenosové kapacity.

Podobný príklad, v ktorom by mohlo byť výhodné použitie multicastu, sú služby využívané veľkým počtom užívateľov. Napríklad internetová televízia, či nejaký server, ponúkajúci napríklad aktuálne správy zobrazujúce sa užívateľom na ich počítačoch. Problémom, ktorý sa tu vyskytuje je najmä to, že v jeden okamžik potrebujeme poslať dáta veľkému množstvu užívateľov, čo značne zaťažuje tak ako vysielaciu stranu, tak prenosovú kapacitu.

Multicastová aplikácia, teda aplikácia, ktorá používa multicast, posiela dáta na multicastovú skupinu. Táto multicastová skupina je množina užívateľov, ktorí majú o dané informácie záujem. Napríklad v spomínanom internetovom vysielaní televízie sú členmi skupiny diváci.

3 Multicast na úrovni 2. a 3. vrstvy

V uvedenej kapitole riešim multicast na úrovni linkovej a sieťovej vrstvy. Zaoberám sa aj protokolmi ARP, IP ale predovšetkým protokolom IGMP. Ďalšou popisovanou záležitosťou je mapovanie MAC adries na multicastové skupiny. V neposlednom rade približujem IGMP snooping.

3.1 Linková vrstva

Linková vrstva je druhou vrstvou ISO/OSI modelu. Zaisťuje prístup k zdieľanému médiu a adresáciu na fyzickom spojení.

Na tejto vrstve sieťová karta na LAN segmente prijíma len pakety určené pre ňu samotnú na základe korešpondujúcej MAC adresy, alebo broadcast MAC adresy. [1]

3.1.1 Protokol ARP (Address Resolution Protocol)

ARP je protokolom siete TCP/IP. Tento protokol zabezpečuje priradenie IP adries fyzickým adresám linkovej vrstvy.

Používa sa v nasledujúcich prípadoch dvoch komunikujúcich zariadení:

- dve zariadenia sú v rovnakých sieťach a jeden chce poslať paket druhému,
- dve zariadenia sú v rôznych sieťach a musia použiť gateway/router k dosiahnutiu druhého zariadenia,
- router potrebuje poslať paket zariadeniu prostredníctvom iného routra,
- router potrebuje poslať paket z jedného zariadenia k cieľu v rovnakej sieti. [2]

Mechanizmus zisťovania MAC adresy pomocou ARP:

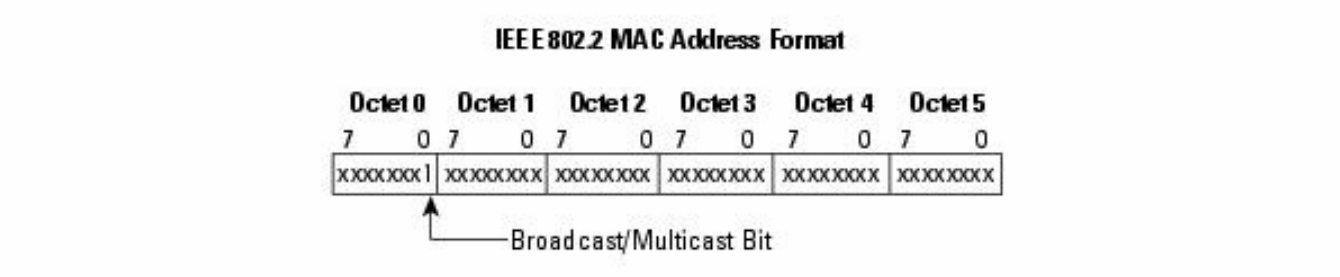
Stanica si v prvej fáze skontroluje, či má v ARP Cache MAC adresu danej IP adresy. Ak nie, musí MAC adresu zistiť.

- stanica A odosiela *ARP Request*, zdrojovou adresou je jej MAC adresa a adresou určenia je broadcast, zdrojová adresa a aj adresa určenia protokolu IP odpovedajú konkrétnym hodnotám,
- všetky uzly sa musia broadcastu venovať a porovnať svoju IP adresu s adresou určenia, ten uzol, ktorý má IP adresu zhodnú s adresou z požiadavky, posiela tzv. *ARP Response*, teda odpoveď s vyplnenou svojou MAC adresou. Ďalšia vzájomná komunikácia oboch uzlov už prebieha pomocou unicastov. [3]

Okrem tohoto existujú ešte špeciálne MAC adresy. Bežné sieťové karty potom majú schopnosť podľa svojho okamžitého nastavenia, filtrovať pakety skupinového vysielania. Najbližším vrstvám programového vybavenia potom už len poskytujú relevantné časti paketov skupinového vysielania,

ktoré sa v lokálnej sieti pohybujú. Ide len o skupiny, ktoré sú predmetom momentálneho záujmu danej stanice. [4]

Štandard IEEE 802.3 umožňuje rozlišovať rámce, ktoré sú určené multicastovej skupine podľa najnižšieho bitu nultého oktetu, tak ako ukazuje obrázok 1. [5]



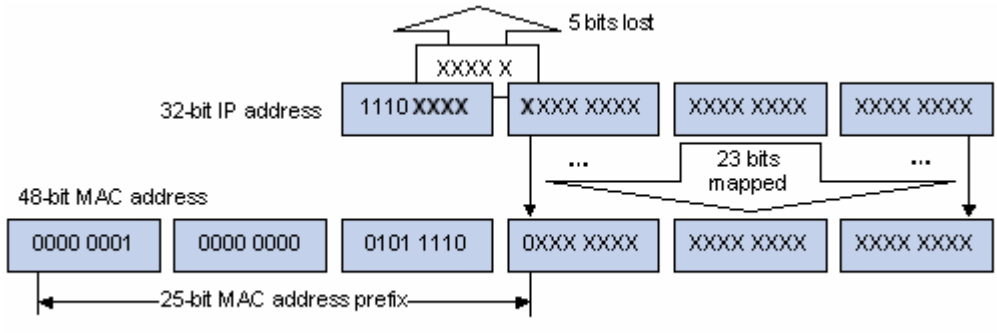
Obrázok 1: Ukážka najnižšieho bitu nultého oktetu multicastového rámca.

3.1.2 Mapovanie MAC adries na multicastové skupiny

Adresy triedy D sú mapované na MAC adresy iným spôsobom, než je tomu pri triedach A, B, alebo C unicastových adries. Unicastové adresy sa viažu s príslušnou MAC adresou priradením IP adresy explicitne k danému zariadeniu, ktoré je potom viazané s korešpondujúcou MAC adresou, alebo dynamickým pridelením. [6]

Na rozdiel od spôsobu mapovania v unicaste, triedy D adries sú automaticky mapované na MAC multicastové adresy jednoduchou procedúrou vysvetlenou nižšie. [6]

MAC adresy sú 48 bitov dlhé, na rozdiel od IP, ktoré majú 32 bitov. Najnižšie bity od 23 bitu IP adresy triedy D sú jednoducho namapované na najnižších 23 bitov MAC multicastovej adresy, tak ako ukazuje obrázok 2. [7]



Obrázok 2: Mapovanie IP adresy triedy D na MAC.

Multicastový rámec začína 24 bitovou hodnotou v hexadecimálnom zápise 01:00:5E. Z tohto teda vyplýva, že 5 bitov z IP adresy sa z procesu mapovania úplne vytratí, a preto nie sú výsledné adresy jednoznačné. Do jednej MAC adresy sa mapuje 2 ^ 5 = 32 IP adries.

Nevýhodou tohoto systému je, že stanica nevie na druhej vrstve určiť, či je rámec pre ňu samotnú, alebo nie je.

3.2 Sieťová vrstva

Na sieťovej vrstve sa do procesu vysielania multicastu zapájajú aj routre. Routre majú dve úlohy:

- získať informácie o tom, ktoré zariadenia - pripojené k routru - majú záujem prijímať multicastový traffic,
- zaistiť posielanie paketov multicastového vysielania do sietí, v ktorých sú záujemci.

3.2.1 Adresy používané v IP multicaste

Než uvediem tabuľku používaných IP adries v multicaste, načrtnem tabuľku tried IP adries.

Triedy IP adries	**Počet bitov sieť/stanica**	**Prvý oktet IP adresy**
trieda A	8 / 24	**0** x x x x x x x
trieda B	16 /16	**1 0** x x x x x x
trieda C	24 / 8	**1 1 0** x x x x x
trieda D	32 / -	**1 1 1 0** x x x x

Tabuľka 1: Triedy IP adries.

Tak, ako ukazuje tabuľka 1. pre adresáciu IP multicastových skupín sa používajú adresy binárne začínajúce **1110**, teda decimálne vyjadrené 224.0.0.0 – 239.255.255.255.

V tabuľke 2. sú uvedené niektoré ďalšie rezervované multicastové adresy. [7]

adresa	**popis**
224.0.0.1	všetky multicastové stroje v danej lokálnej sieti (stanice aj routre)
224.0.0.2	všetky multicastové routre v lokálnej sieti
224.0.0.4	všetky DVMRP routre
224.0.0.5	všetky OSPF routre
224.0.0.6	všetky Designated OSPF routre
224.0.0.7	všetky Shared Tree routre
224.0.0.9	všetky RIP2 routre
224.0.0.13	všetky PIM routre
224.0.0.15	všetky CBT routre
224.0.1.40	všetky Cisco PIM routre

Tabuľka 2: Niektoré rezervované multicastové adresy.

3.2.2 Protokol IGMP

Router, ktorý príjme multicastové dáta určené pre konkrétnu multicastovú skupinu musí vedieť, na ktoré zo svojich rozhraní má dáta posielať. Práve o túto úlohu routra sa stará protokol IGMP. Existuje niekoľko verzií IGMP, ako definuje RFC dokument. IGMP v1 je definovaný v RFC 1112, IGMP v2 v RFC 2236 a IGMP v3 v RFC 3376.

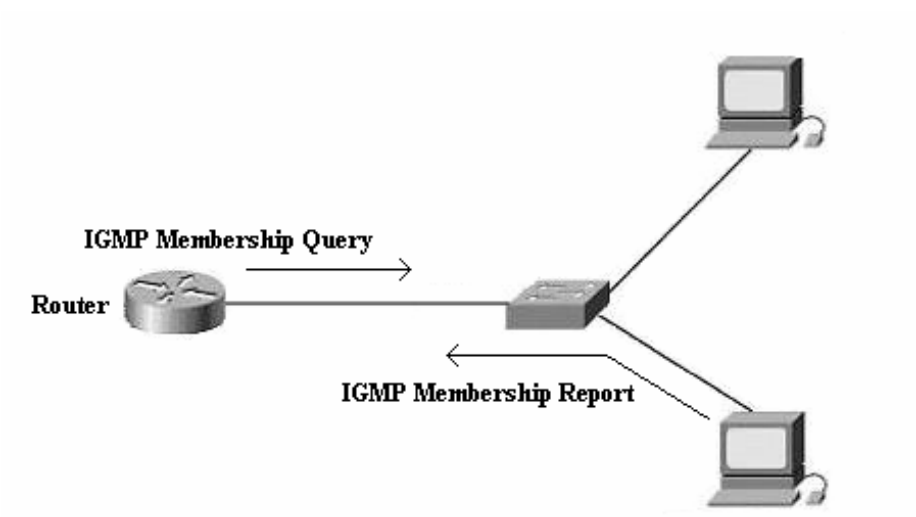
3.2.2.1 IGMP v1

IGMP v1 bol kedysi často používaný, ale v súčasnosti je považovaný za zastaralý protokol. V IGMP v1 sú k dispozícii tieto dva typy správ:

- *Membership Query*
- *Membership Report*

Multicastové routre posielajú správu typu *Membership Query* k zisteniu, či má daná stanica ešte záujem prijímať správy pre danú skupinu. Tieto dotazy sú adresované všetkým zariadeniam a nesú v IP hodnotu TTL nastavenú na 1.

Príjemcovia odpovedajú na *Membership Query* generovaním *Membership Report*. Takto odpovedá každé zariadenie danej skupiny, do ktorej patrí na sieťovom rozhraní, z ktorého prišiel daný *Membership Query*. Situácia je znázornená na obrázku 3. Jeden z počítačov odpovedá na dotaz routru.



Obrázok 3: Znázornená situácia vysielania dotazu a odpovede.

Pretože je bežné, že skupina má viac staníc pre prijímanie dát, za účelom nezahltenia routra konkurenčnými *Membership Report* správami a redukciou ich celkového počtu, používajú sa dve techniky:

- Keď host príjme *Membership Query*, namiesto toho, aby hneď posielal *Membership Report*, zvolí si pre každú skupinu, náhodné číslo pre časovač v rozsahu 0 až D sekúnd.

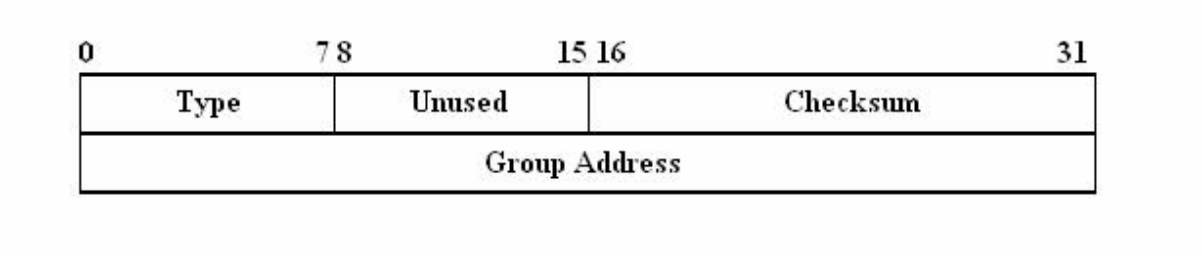
Toto číslo vyjadruje čas v sekundách, za ktorý chce poslať *Membership Report*. Keď tento časovač vyprší, generuje sa *Membership Report* pre korešpondujúcu skupinu.

- *Membership Report* je poslaný s IP adresou cieľovej skupiny a TTL nastaveným na 1, takže ostatní členovia rovnakej skupiny v tej istej sieti môžu počuť tento *Membership Report*. Keď sa tak stane, stanica zastaví svoj časovač pre danú skupinu a negeneruje žiaden *Membership Report* pre túto skupinu. Týmto sa za bežných okolností vyšle len jeden *Membership Report* pre každú skupinu na danej sieti členom skupiny s časovačom, ktorý vyprší ako prvý. Routre nepotrebujú vedieť, ktoré konkrétne stanice patria ku skupine, stačí aby boli informované, že aspoň jedna stanica patrí ku skupine na príslušnom rozhraní. Poznamenávam, že multicastové routre prijímajú všetky IP multicastové pakety, preto nemusia byť adresované explicitne.

V prípade, že klient danú skupinu opustí, prestane odpovedať na *Membership Query*. Ak router pre danú skupinu nedostane tri krát po sebe žiadny *Membership Report*, prestane túto správu posielať. Tento jednoduchý mechanizmus je ale nevýhodný v prípade, že stanice často menia členstvo v skupinách a tým môžu sieť preťažovať. [8]

Obrázok 4. ukazuje formát IGMP v1 správy, pričom:

- **Type** – obsahuje číslo udávajúce typ správy (*Membership Query* alebo *Membership Report*),
- **Unused -** táto položka je vo verzii 1 protokolu IGMP nevyužitá,
- **Checksum -** položka, zaisťujúca kontrolný súčet. Pri výpočte je vynulovaná,
- **Group Address** – pri správe typu *Membership Query* je toto pole vynulované, pri *Membership Report* je v tomto políčku adresa skupiny, do ktorej chce byť člen pripojený.



Obrázok 4: Formát IGMP v1 správy.

3.2.2.2 IGMP v2

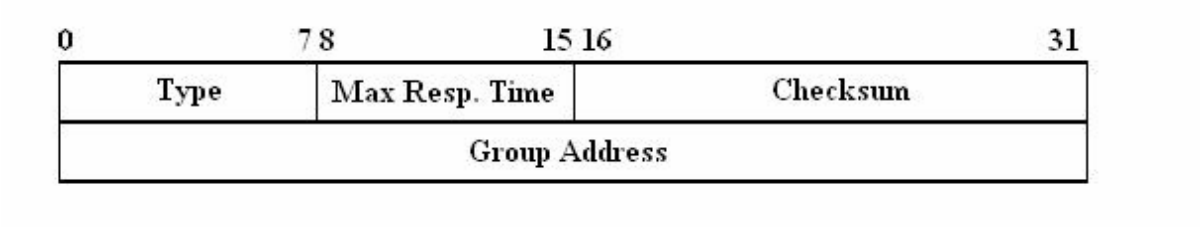
Hlavný rozdiel IGMP v2 od v1 je v tom, že oproti prvej verzii obsahuje ďalší typ správy *Leave Group*. Keď stanica opustí danú multicastovú skupinu, táto môže aktívne vyjadriť svoj úmysel pomocou tejto správy. Router potom pomocou správy typu *Membership Query* zistí, či sa ešte vyskytujú nejakí klienti danej skupiny. Pokiaľ nedostane žiadnu odpoveď, prestane posielať dáta pre

danú skupinu, ak je ale ešte niekto členom skupiny a odpovie, router pokračuje ďalej v posielaní. Týmto sa zníži latencia pri odhlasovaní zo skupiny a aj sieťový traffic celkovo.

V prípade, že na danej sieti je niekoľko multicastových routrov, IGMP v1 požaduje aby multicastový smerovací protokol rozhodol, ktorý router bude vybraný k posielaniu IGMP *Membership Query* dotazov. V IGMP v2, multicastový router s najnižšou IP adresou v sieti je automaticky vybraný k tejto činnosti. [6]

IGMP v2 je spätne kompatibilný s IGMP v1. Ak stanice detekujú IGMP v1 router podľa jeho *Membership Query* správ, tiež začnú posielať správy v IGMP v1 a časovač si nastavia na 400 sekúnd. Keď daný časový interval uplynie, začnú opäť posielať IGMP v2. Naopak, router je schopný akceptovať stanice s IGMP v1 aj s IGMP v2, pretože dokáže ich správy obslúžiť. Ak router detekuje IGMP v1 zariadenia na sieti, ignoruje správy typu *Leave Group*. [9]

Formát IGMP v2 je na obrázku 5.



Obrázok 5: Formát IGMP v2 správy.

Formát IGMP v2 je veľmi podobný IGMP v1:

- **Type** – obsahuje číslo udávajúce typ správy:
  0x11 - *Membership Query,*
  0x16 - M*embership Report,*
  0x17 – *Leave Group*.
- **Max Response Time –** určuje maximálnu povolenú dobu pre zasielanie odpovedí pri správe typu *Membership Query*,
- **Checksum -** políčko, zaisťujúce kontrolný súčet. Pri výpočte je vynulované,
- **Group Address** – pri správe typu *Membership Query* je toto pole vynulované, pri *Membership Report* a *Leave Group* je v tomto políčku adresa skupiny, do ktorej chce byť člen pripojený.

3.2.2.3 IGMP v3

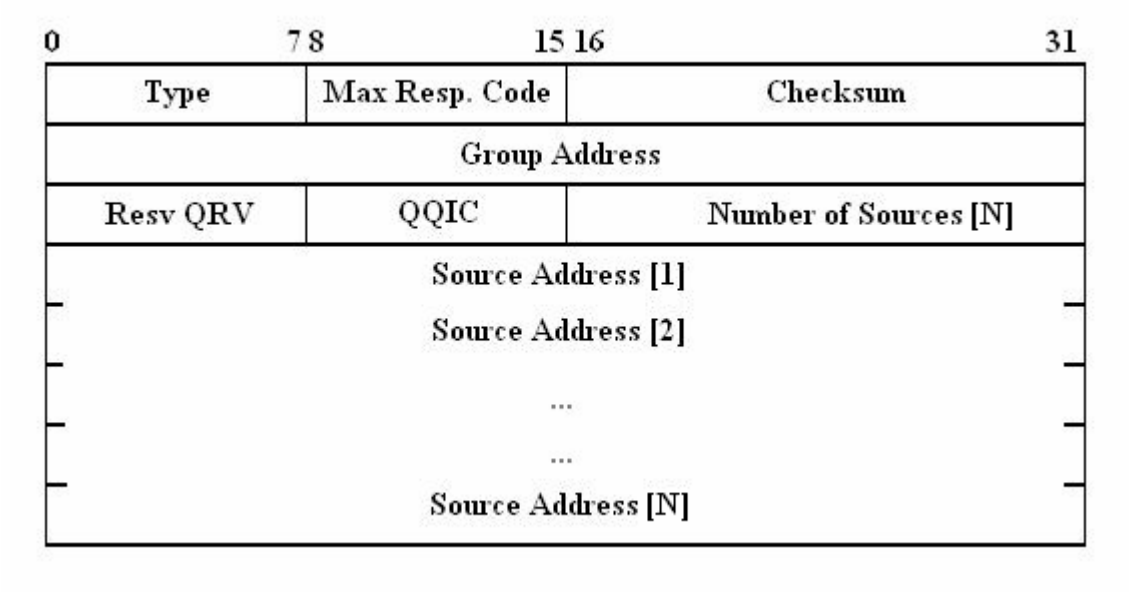
Primárna funkcia pridaná v IGMP v3 je vlastnosť stanice určiť si len vybrané zdroje z nejakej multicastovej skupiny, o ktorých dáta prejaví záujem, nie implicitne od všetkých, ako je tomu v predchádzajúcich typoch IGMP protokolu. Použitím protokolu IGMP v3 sa preto klienti neprihlasujú len do skupiny, ale aj k odberu dát od konkrétneho zdroja v skupine. Táto vlastnosť dáva príjemcom

väčšiu možnosť kontroly nad príjmom dát od zdrojov skupiny. Spomínaná vlastnosť je uskutočňovaná prostredníctvom dvoch typov správ:

- *Inclusion Group-Source Report* správa dovoľuje špecifikovať IP adresu zdroja, z ktorého chce stanica prijímať,
- *Exclusion Group-Source Report* správa dovoľuje stanici špecifikovať zdroje, z ktorých nechce dáta prijímať.

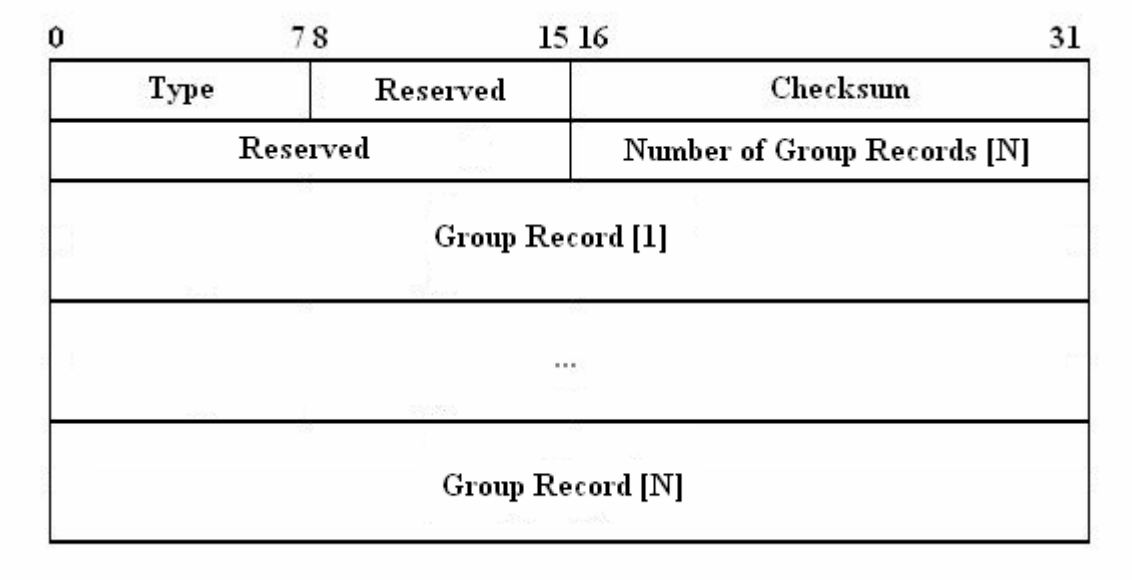
V IGMP v1 a IGMP v2 je celý traffic zo všetkých zdrojov danej skupiny posielaný do siete, v ktorej je nejaký člen. Naviac, *Leave Group* správa bola v IGMP v3 vylepšená oproti IGMP v2. *Group-Source Leave* správa dovoľuje špecifikovať stanici IP adresu zdroja z ktorejkoľvek skupiny, ktorú si praje opustiť.

Membership Query správa je na obrázku 6. [10]



Obrázok 6: Formát IGMP v3 *Membership Query* správy.

Formát správy *Membership Report* na obrázku 7. [10]



Obrázok 7: Formát IGMP v3 Membership Report správy.

Jednotlivé položky znamenajú:

- **Max Response Code –** určuje maximálnu povolenú dobu pre zasielanie odpovedí,
- **Checksum -** políčko, zaisťujúce kontrolný súčet. Pri výpočte je vynulované,
- **Group Address –** pri type správy *Membership Query* je nastavené na nulu, pri ostatných dotazoch nastavené na IP multicastovú adresu, ktorá je dotazovaná,
- **Resv –** nastavené na nulu,
- **QRV –** nenulová hodnota je použitá dotazovačom,
- **QQIC –** špecifikácia intervalu pre odpoveď od zariadenia, ktoré sa dotazuje,
- **Number of Sources [N]** – políčko obsahuje počet zdrojových adries prezentovaných v dotaze.
- **Source Address** – obsahuje IP unicastové adresy.

V praktickej časti diplomovej práce sa zaoberám tvorbou IGMP paketu na nižšej úrovni. Tzn. že sa zaoberám taktiež tvorbou IP protokolu, resp. plnením políčok tohoto protokolu.

So switchom a protokolom IGMP súvisí jeden pojem, ktorý testujem v praktickej časti, preto uvediem potrebné teoretické informácie.

3.2.2.4 IGMP snooping

Pojem IGMP snooping súvisi so switchom. Switch je zariadenie, pracujúce na 2. vrstve, teda linkovej vrstve ISO/OSI modelu. Z tohoto dôvodu, všetok multicastový traffic, ktorý príjme na niektorom zo svojich rozhraní vysiela na všetky ostatné rozhrania, bez ohľadu nato, či na danom rozhraní niekto o tieto dáta má záujem. Daná skutočnosť môže výrazne znižovať efektivitu multicastu. Preto, niektoré switche obsahujú vlastnosť IGMP snooping.

Táto technika skúma obsah multicastového trafficu. Switch, ktorý má túto schopnosť, dokáže rozumieť paketom na 3. vrstve ISO/OSI modelu. Keď switch príjme správu *Membership Report*,

priradí si záznam o príslušnom porte do svojej pamäti a začne posielať dáta pre danú multicastovú adresu. Switch ale v tomto momente neprepošle *Membership Report* späť do siete. Prinúti na *Membership Query* odpovedať aj ostatných záujemcov, čím zistí, či sú na sieti ešte aj iní príjemci.

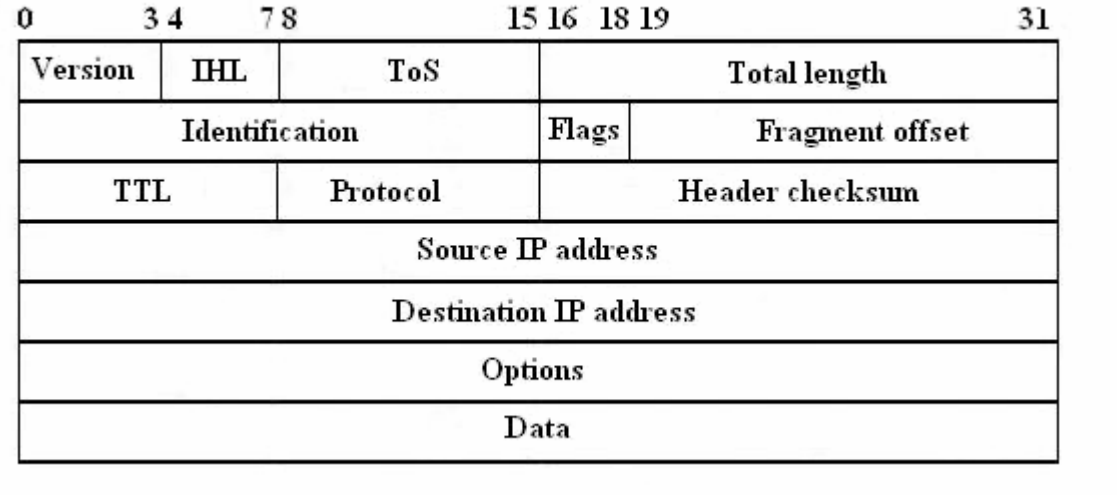
V situácii, kedy sa niektorý z prihlásených snaží odhlásiť, pošle *Leave Group* správu. Nato však switch reaguje vyslaním *Membership Query* správou, pretože si nemôže byť istý, že na danom rozhraní neboli aj iní záujemci.

V bežnej prevádzke posiela dotaz *Membership Query* len jeden z routrov. Switch každú správu prepošle na všetky porty a odpovede si necháva pre seba.

Switch však potrebuje vedieť, na ktorých portoch má pripojené routre, preto skúma aj obsah ostatných paketov, predovšetkým PIM a DVMRP, ktoré by mu signalizovali router. [11]

3.2.2.5 IP protokol

IP protokol je komunikačný protokol, na ktorom je dnes postavený Internet. Zaisťuje komunikáciu dvoch počítačov. Komunikácia prebieha výmenou IP paketov. Formát IP paketu je na obrázku 8.



Obrázok 8: Formát IP paketu. [12]

Jednotlivé položky protokolu IP znamenajú nasledovné:

- **Version** - verzia IP protokolu, v tejto práci je použitá hodnota 4 ako IPv4,
- **IHL** - dĺžka IP záhlavia, ktoré môže obsahovať nepovinné položky, jeho veľkosť preto nie je daná dopredu,
- **ToS** - typ služby,
- **Total length** - celková dĺžka, tj. hlavička a dáta,
- **Identification** - identifikácia, toto číslo prideľuje operačný systém, súvisí s pojmom fragmentácie,
- **Flags** - príznaky, ktorými môžeme ovplyvniť fragmentovanie,

- **Offset fragment** - posunutie fragmentu v pôvodnom IP pakete,
- **TTL** (Time To Live) - číslo, ktoré udáva životnosť paketu, presnejšie to znamená, koľkými routrami môže paket prejsť. V každom smerovači je táto hodnota znížená o 1 a v prípade, že dosiahne 0, paket je zahodený,
- **Protocol** - identifikácia protokolu - udáva, aké dáta bude paket prenášať,
- **Checksum** - kontrolný súčet,
- **Source Address** - zdrojová adresa (32 bitov),
- **Destination Address** - cieľová adresa (32 bitov),
- **Options** - ďalšie nastavenia,
- **Dáta** - dáta IP protokolu.

4 Multicastové smerovacie protokoly

Smerovanie pre multicastové pakety je potrebné uskutočňovať trochu inak, než v klasickom unicastovom smerovaní IP paketov. Vzhľadom k tomu v tejto kapitole rozoberiem tie multicastové smerovacie protokoly, ktoré sú routrami firmy Cisco podporované. Ďalej sa venujem zdrojovým a zdieľaným stromom používaným v multicastových smerovacích protokoloch.

Hlavné rozdiely medzi klasickými unicastovými a multicastovými smerovacími protokolmi sú nasledovné: [13]

- multicastové cesty sa menia tým, ako sa účastníci prihlasujú a odhlasujú. Normálne cesty v unicastovom smerovaní sa menia len so zmenou topológie siete,
- pri smerovaní je treba brať do úvahy okrem cieľovej adresy aj ďalšie informácie, pretože samotná cieľová adresa nestačí.

Routre v sieti používajú multicastové smerovacie protokoly k optimálnej ceste multicastového paketu cez sieť od zdroja k viacerým cieľom, ktorý pozostáva z členov danej multicastovej skupiny.

Niektoré multicastové smerovacie protokoly sú odvodené od unicastových smerovacích protokolov. Tieto unicastové smerovacie protokoly používajú jednu z dvoch techník:

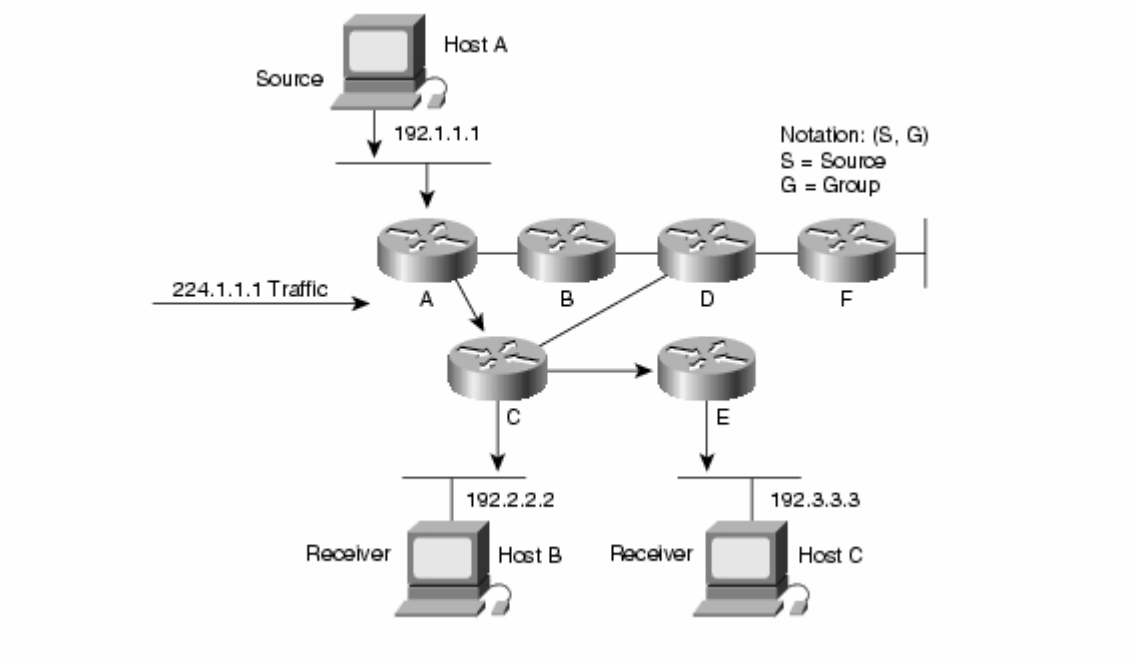
- distance vector – najznámejší reprezentant RIP protokol,
- link state – predstaviteľ OSPF protokol.

Každý multicastový smerovací protokol musí zostaviť distribučný strom k smerovaniu multicastových paketov členom skupiny optimálnou cestou. Existujú dva základne typy týchto stromov:

- zdrojový strom (source tree),
- zdieľaný strom (shared tree).

4.1 Zdrojový strom

Zdrojový strom je najjednoduchší typ distribučného stromu. Niekedy sa mu tiež hovorí strom najkratších ciest (*Shortest Path Tree* - SPT). Zdroj multicastového prenosu je umiestnený v koreni stromu, cieľové stanice sú umiestnené v listoch. Multicastový traffic je prenášaný stromom od zdroja smerom k jednotlivým príjemcom. Rozhodnutie o tom, na ktoré rozhranie bude multicastový paket poslaný závisí na multicastových smerovacích tabuľkách. Pre označenie stromu sa používa notácia (S, G) kde S je adresa zdroja a G je adresa multicastovej skupiny. Príklad zdrojového stromu je na obrázku 9. [1]

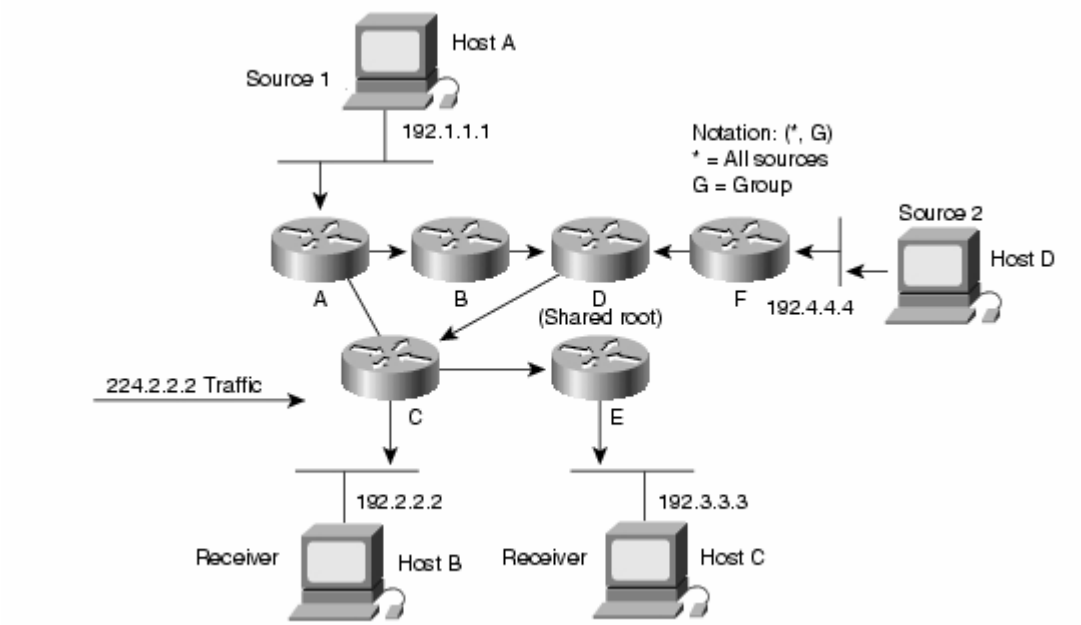


Obrázok 9: Príklad stromu najkratších ciest pre multicastovú skupinu 224.1.1.1 s koreňom v stanici A a pripojenými dvoma príjemcami dát – stanice B a C.

4.2 Zdieľaný strom

Zdieľaný strom sa líši od zdrojového stromu tým, že koreň stromu je vždy na jednom mieste, nezávisle na tom, kto dáta posiela. Tento koreň sa nazýva *Rendezvous Point* (RP). RP je bod, do ktorého musia zdroje preniesť traffic. Keď sa príjemca pridá do skupiny so zdieľaným stromom, koreňom bude vždy RP, a multicastový traffic je prenášaný z tohto bodu RP príjemcom.

Ako označenie pre tieto stromy sa používa notácia (*, G). Hviezdička označuje, že všetky zdroje v príslušnej skupine zdieľajú ten istý strom, tak ako ukazuje obrázok 10. [1]



Obrázok 10: Zdieľaný strom pre skupinu 224.2.2.2 s koreňom umiestneným v routry D. Pri použití zdieľaného stromu, zdroj musí posielať traffic koreňu (RP) a potom je traffic posielaný stromom ostatným príjemcom.

Zdieľané stromy nie sú optimálne v smerovaní tak ako zdrojové stromy, pretože všetok traffic od zdrojov musí putovať k RP a potom k príjemcom. Avšak, množstvo potrebných informácií k smerovaniu je menšie než u zdrojového stromu.

Zdrojové stromy môžu byť rozdelené na jednosmerné a obojsmerné. Jednosmerné stromy sú v zásade doteraz popísané stromy, obojsmerné stromy dovoľujú multicastovému trafficu putovať stromom smerom hore aj dole. Obojsmerné stromy sú užitočné, keď máme mnoho zdrojov a príjemci sú rozptýlení v sieti.

4.3 Dense Mode protokoly

Protokoly tejto skupiny používajú zdrojové stromy a pracujú na tzv. *Push* princípe, čo znamená, že multicastový traffic je primárne prenášaný všade. Aby sa zabránilo zahlteniu zo strany príjemcov, listy stromu, ktoré nemajú žiadnych príjemcov pre danú skupinu, pošlú smerom ku koreňu tzv. *Prune* správu. Smer, z ktorého táto správa prišla, je potom v strome orezaný, čím zostanú len vetvy, ktoré majú nejakých aktívnych príjemcov. *Prune* správa platí len obmedzenú dobu, takže po chvíli je nutné ju opäť obnoviť, inak začne nadradený router dáta opäť posielať. [14]

Predstaviteľmi Dense Mode protokolov sú napríklad protokoly DVMRP a PIM - DM.

4.3.1 DVMRP (Distance Vector Multicast Routing Protocol)

DVMRP je jedným z najstarších protokolov, ktorý navrhol Steve Deering. Stavia na princípe unicastového distance-vector smerovacieho protokolu RIP. Hlavný rozdiel je v tom, že RIP ma za úlohu smerovať pakety k určitému cieľu, DVMRP naopak sleduje cesty naspäť k zdroju multicastových paketov a týmto sa môže skonštruovať zdrojový strom. Multicastový algoritmus vyžaduje vytvorenie stromov na základe smerovacích informácií. Tento proces tvorby stromu vyžaduje viac informácií než poskytuje RIP, preto je DVMRP oveľa komplikovanejší. [15]

DVMRP používa techniku známu ako *Reverse Path Multicasting* (RPM). Keď router príjme paket na niektorom zo svojich rozhraní, pomocou *Reverse Path* testu skontroluje, či to je najkratšia cesta ku zdroju použitím unicastovej smerovacej tabuľky. Ak paket príde z najkratšej cesty od zdroja multicastového trafficu, router pošle paket na všetky svoje rozhrania, okrem toho, ktoré vedie späť k zdroju paketu. V inom prípade je paket zahodený, pretože neprichádza z optimálnej cesty a môže to spôsobiť redundanciu paketu. [6]

DVMRP používa unicastový smerovací protokol, ktorý určí najkratšiu cestu k zdroju. Táto smerovacia informácia je vymieňaná medzi DVMRP routrami v sieti.

Ak je router pripojený k sieti, v ktorej už nie je žiadna stanica, ktorá by chcela prijímať multicastové dáta, pošle *Prune* správu. Tento princíp zabraňuje posielaniu paketov do siete, kde nie je žiadny príjemca danej multicastovej skupiny. Podobne, G*raft* správa sa používa k dynamickému prihlasovaniu členov ku skupine. [6]

4.3.2 PIM - DM (Protocol Independent Multicast – Dense Mode)

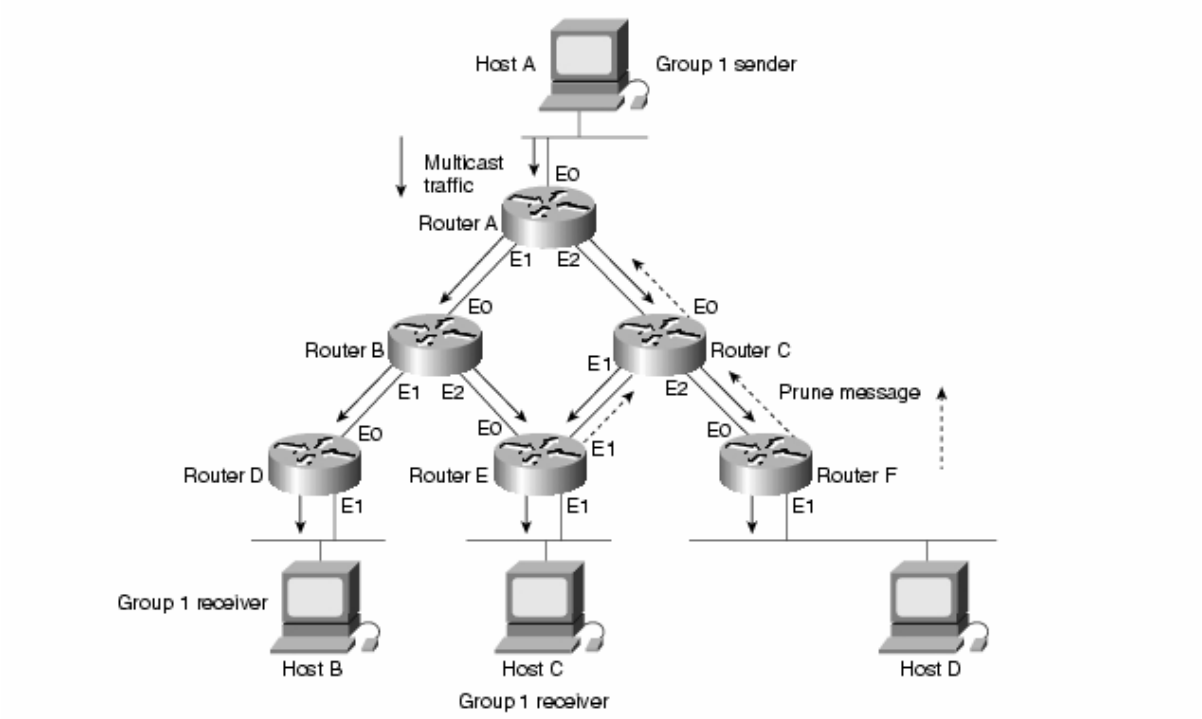
Protocol Independent Multicast má dva módy, Dense Mode a Sparse Mode. Primárne bol vytvorený pre Sparse Mode, ktorého úlohou je zvýšiť efektivitu multicastových smerovacích protokolov medzi WAN.

PIM – DM pracuje podobne ako DVMRP, predpokladá, že keď zdroj začne posielať dáta, všetky stanice chcú prijímať multicastové pakety. Na začiatku, multicastové pakety sú posielané všetkým staniciam v sieti. PIM – DM používa RPF k ochrane pred redundanciou paketov. Ak niektoré časti siete nemajú členov skupiny, PIM – DM odstráni túto vetvu stromu. Tento *Prune* stav má nejaký časový limit, po ktorom sú dáta opäť posielané aj do vetiev stromu odrezaných v predošlom kroku. [16]

Jeden veľký rozdiel oproti DVMRP je ten, že PIM – DM závisí na existujúcom unicastovom smerovacom protokole. Používa sa k určeniu ciest späť ku zdroju, naproti tomu DVMRP používa vlastný protokol. [6]

K minimalizácii opakovaného posielania paketov a nasledujúceho odrezania, PIM – DM používa *State Refresh* správu. Táto správa je posielaná routrom priamo pripojeným k zdroju a je šírená sieťou. Keď je prijatá routrom na jeho RPF rozhraní, *State Refresh* správa znamená, že daný *Prune* stav bude aktualizovaný.

Obrázok 11. ukazuje ako pracuje PIM – DM. [17] Znázorňuje situáciu, kedy router A prijíma multicastový traffic od stanice A na rozhraní Ethernet E0, zduplikuje každý paket, a pošle na rozhranie Ethernet E1 a Ethernet E2 routru B a C. Tieto opäť zduplikujú pakety a pošlú routrom D, E a F. Router D má priamo pripojeného príjemcu, ktorý je členom skupiny 1, takže router D nepošle *Prune* správu. Podobne router E, ale pretože router E príjme paket na dvoch rozhraniach pošle *Prune* správu routru C. Router F nemá žiadnych členov skupiny 1, preto pošle *Prune* správu routru C. Tento pošle *Prune* správu routru A a týmto router A bude posielať multicastový traffic pre skupinu 1 len routru B.



Obrázok 11: Znázornený princíp protokolu PIM – DM.

4.3.2.1 Hello správy

PIM – DM používa *Hello* správy k detekcii ostatných PIM routrov. *Hello* správy sú posielané periodicky na každé rozhranie, na ktorom je povolený PIM protokol. Táto perióda je raz za 30 sekúnd. *Hello* správy sú multicastované do skupiny na všetky PIM routre. Keď je PIM povolený na rozhraní alebo je router práve spustený, *Hello* časovač musí byť nastavený na nejaké náhodné číslo v rozmedzí 0 až *Triggered_Hello_Delay*. Týmto zamedzíme synchronizácii týchto správ v prípade, že viaceré routre boli spustené naraz.

Po počiatočnej *Hello* správe, musí byť táto správa posielaná každú *Hello_Period*.

4.3.2.2 Prijímanie Hello správ

Keď router príjme *Hello* správu, zaznamená si rozhranie, z ktorého túto správu prijal. Ďalej si zaznamená router, ktorý túto správu poslal a ďalšie dodatočné informácie. Tieto informácie sú ponechané niekoľko sekúnd v *Hold_Time* políčku *Hello* správy. Ak je prijatá nová *Hello* správa od príslušného susedného routra, *Neighbor_Liveness_Timer* musí byť nastavený na nové *Holdtime* prijatej *Hello* správy.

4.4 Sparse Mode protokoly

Tieto protokoly používajú zdieľané stromy pre distribúciu multicastových dát a využívajú tzv. *Pull* model. Tento model predpokladá, že dáta sa nesmú posielať do siete, pokiaľ si ich niekto explicitne nevyžiada. Pokiaľ by sa niektorá stanica chcela pripojiť do multicastovej skupiny, jeho príslušný router pošle správu *Join* smerom ku koreňu stromu. Takto bude zostavená ďalšia vetva stromu a dáta môžu prúdiť. Pokiaľ už v danej vetve nebude žiaden príjemca skupiny, router pošle správu *Graft*, ktorá vetvu odreže. [14]

Medzi tieto protokoly patria napríklad PIM - SM (Protocol Independent Multicast) a CBT (Core Base Tree).

4.4.1 PIM - SM (Protocol Independent Multicast – Sparse Mode)

V protokole PIM – SM sa členovia skupiny prihlasujú k RP. Routre prijímajú explicitne *Join/Prune* správy od susedných routrov, ktoré majú členov skupiny po prúde stromom. Router potom posiela dáta adresované do multicastovej skupiny, len na to rozhranie, na ktorom bola explicitne prijatá *Join* správa.

PIM – SM router s najvyššou IP adresou je vybraný ako tzv. *Designated Router* (DR) v danej podsieti. Keď príjemca vyjadrí záujem prijímať dáta z určitej multicastovej skupiny, oznámi to DR routru (jeden z routrov, ku ktorým je príjemca priamo pripojený) prostredníctvom IGMP. Tento DR periodicky posiela *Join/Prune* správy smerom k RP pre každú skupinu, v ktorej má aktívnych členov. DR router určuje RP, ktorý je k skupine pridelený a posiela unicastovo *Join* správu k RP danej skupiny. Táto správa sa propaguje až k RP, alebo k routru, ktorý už je na strom pripojený. Na každom smerovači po ceste sa vytvorí zodpovedajúca stavová informácia a dáta začnú tiecť z RP k príjemcom. Periodicky je potrebné posielať *Join* správy, pretože inak by bola vetva po určitom čase zo stromu odrezaná.

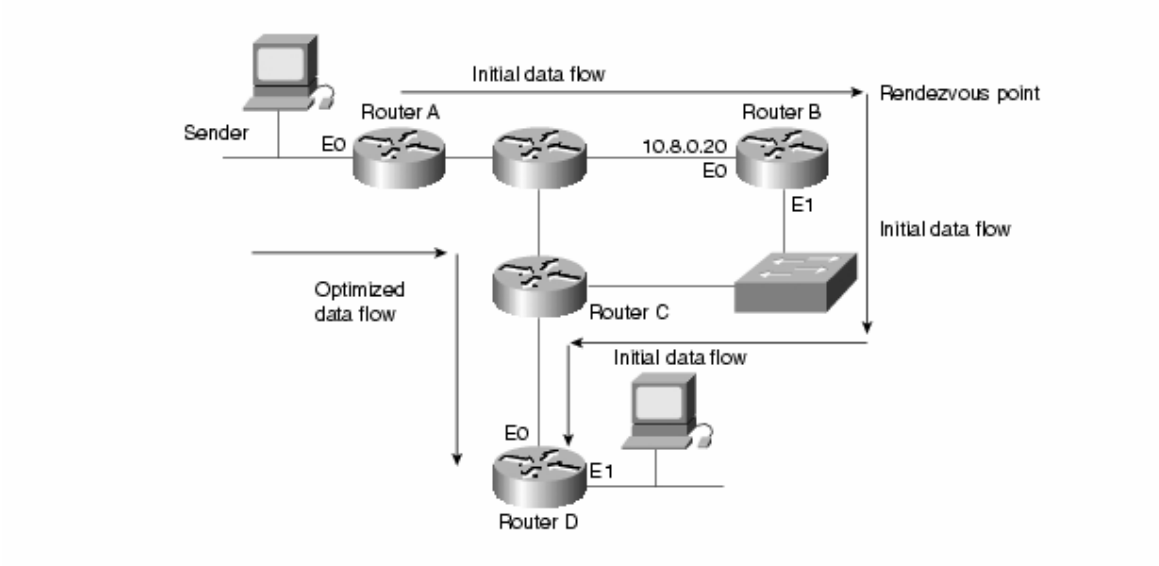
Zdroju dát stačí len poslať dáta s adresou požadovanej skupiny. Jeho príslušný DR tieto dáta zabalí a unicastovo v *Register* správe pošle RP, ktorý ich po odbalení pošle po vytvorenom zdieľanom strome. Aby sa routre vyhli tomuto zabaľovaniu a odbaľovaniu a zabránili tým zaťaženiu

routra, RP po prijatí týchto unicastových dát pošle správu *Join* priamo k DR zdroju dát za účelom vytvorenia vetvy stromu SPT. Akonáhle je vytvorený SPT od zdrojového routra po RP, multicastový traffic začne prúdiť od zdroja k RP už bez zapúzdrovania. Toto pripojí zdroj k RP zdrojovým stromom, ktorý po prijatí správy prestane dáta posielať unicastovo. Keď RP príjme multicastové dáta od zdroja pošle *Register Stop* správu zdroju za účelom oznámenia routru aby prestal posielať unicastové *Register* správy. V tomto bode, multicastový traffic prúdi od zdroja prostredníctvom SPT k RP a od tohto miesta zdieľaným stromom k príjemcom. [18]

PIM – SM má tú vlastnosť, že routre, ktoré majú priamo pripojených členov sa môžu rozhodnúť, že dáta budú doručovať kratšou cestou, než cez RP. Tým vlastne vynechať RP. Väčšinou sa tak deje v závislosti na prevádzke. Členovia skupiny odchod zo skupiny oznamujú tým, že DR pošle *PIM Prune* správu RP a tým je táto časť stromu odrezaná.

Existovať môže niekoľko rôznych RP pre rôzne multicastové skupiny, ale bežne len jeden RP pre skupinu a najjednoduchšia implementácia používa len jeden RP pre všetky multicastové skupiny. Princíp protokolu PIM – SM ukazuje obrázok 12. [17]

Na obrázku 12. sú taktiež znázornené routre A a D ako listové routre, tzn. routre, ktoré sú priamo pripojené k príjemcovi alebo odosielateľovi multicastových dát. Sparse Mode konfigurácia určuje jeden alebo niekoľko routrov ako RP. V našom príklade – router B. Listový router, ktorý je priamo pripojený k odosielateľovi (router A) posiela *PIM Register* správy odosielateľa k RP. Listový router, ktorý je priamo pripojený k príjemcovi (router B) posiela *PIM Join* a *Prune* správy k RP za účelom informovať ho ohľadom svojho členstva v skupine.



Obrázok 12: Vysvetľuje prácu PIM – SM.

Mechanizmy PIM – SM protokolu:

- *zisťovanie susedov – PIM Hello* správy sú periodicky posielané za účelom zistenia existencie ďalších PIM routrov na sieti a k voľbe DR,
- *stavy* – popisuje stavy multicastových distribučných stromov,
- *prihlasovanie* – listové routre sa pripájajú k zdieľaným stromom. Keď router chce prijímať multicastový traffic pre danú skupinu, pošle *PIM Join* správu k RP,
- *registrácia* – router zapúzdruje multicastový paket do *PIM Register* správy a pošle unicatovo ku RP, ktorý tieto *Register* správy od routrov prijíma,
- *SPT prepínanie* – umožňuje prepínať v PIM - SM konfiguráciu na SPT,
- *odrezávanie* – stanice, ktoré opúšťajú skupinu používajú *Prune* správu.

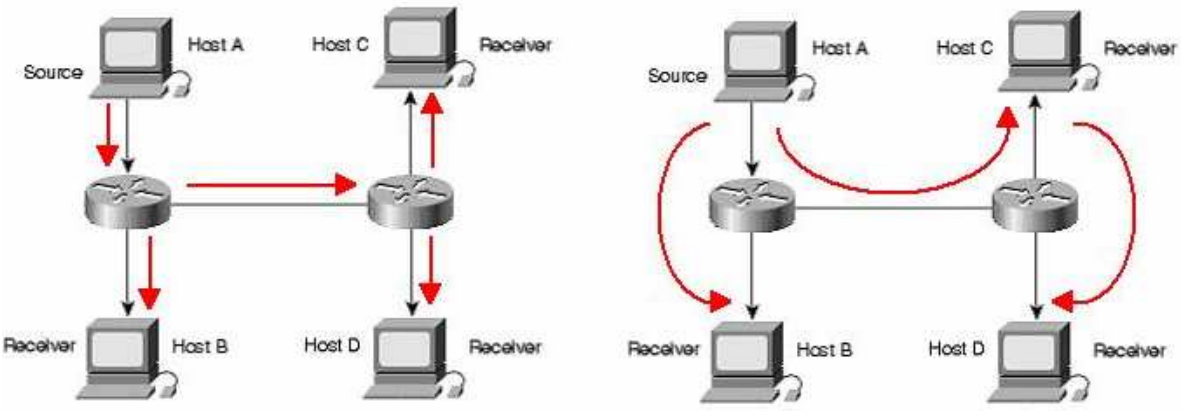
5 Implementácia

V tejto kapitole by som rád priblížil možnosti implementácie multicastu, ktoré sa ponúkajú a programy, ktoré som vytvoril. Tieto programy využívajú BSD socketov, ktoré taktiež v diplomovej práci popisujem.

Ako už bolo v úvode práce sčasti naznačené, existujú tri typy implementácie multicastu:

One to all unicast – už spomenutý spôsob doručovania paketov záujemcom typickým unicastovým spôsobom, teda takým, že vysielač dát odosiela každému z príjemcov dáta unicastovo. Tento spôsob výrazne zaťažuje vysielač dát.

Application level multicast – dôvod existencie tohoto spôsobu implementácie multicastu je jednoduchý. Rozšírenie network - layer multicastu (inými slovami explicitného multicastu – bude popísaný neskôr) nie je ešte príliš podporované poskytovateľmi Internetu. Z tohto dôvodu veľká časť Internetu ešte stále nie je schopná poskytovať potrebné prostriedky pre používanie multicastu. Základná myšlienka aplikačného multicastu je vysvetlená na obrázku 13. Na rozdiel od bežného multicastu, kde dáta sú replikované multicastovými routrami, v aplikačnom multicaste dáta sú replikované na koncových staniciach. S týmto sa vynára otázka, ako budú vedieť jednotlivé stanice, kde dáta ďalej posielať, aby nedochádzalo k nepríjemným javom akým je napríklad redundancia dát. Taktiež v aplikačnom multicaste sa vyskytujú protokoly – aplikačné multicastové protokoly, ktorých úlohou je práve toto určovanie. Teda komu následne má koncová stanica, ktorá získala dáta, ďalej preposlať. Ako znázorňuje obrázok, na rozdiel od bežného multicastu, v aplikačnom multicaste sa vyskytujú situácie, kedy musí byť paket poslaný tou istou linkou, ktorou prišiel. Táto skutočnosť robí aplikačný multicast menej výkonný ako explicitný multicast. [19]



Obrázok 13: Na ľavej strane network - layer multicast na pravej application layer multicast [19].

Explicitný multicast – pod týmto pojmom si môžeme predstaviť klasický multicast. Implementovať explicitný multicast predpokladá podporu zo sieťovej vrstvy. Konkrétne to znamená

podporu zo strany routrov, ktoré musia umožňovať preposielať multicastový traffic a taktiež podporovať multicastové protokoly.

Pri spracovávaní praktickej časti diplomovej práce som mal k dispozícii prístup do laboratória. Nachádzajú sa tu routre podporujúce multicast a taktiež ostatné zariadenia potrebné k otestovaniu tohoto typu multicastu. Testy, ktoré som uskutočňoval, následne popísal a zhodnotil v ďalšej kapitole sa preto týkajú explicitného multicastu.

Samotné testovanie multicastu sa nezaobíde bez implementovania programu. Ako prostriedok využitý pri testovaní som sa rozhodol použiť BSD sockety, keďže som sa s nimi stretol už počas štúdia.

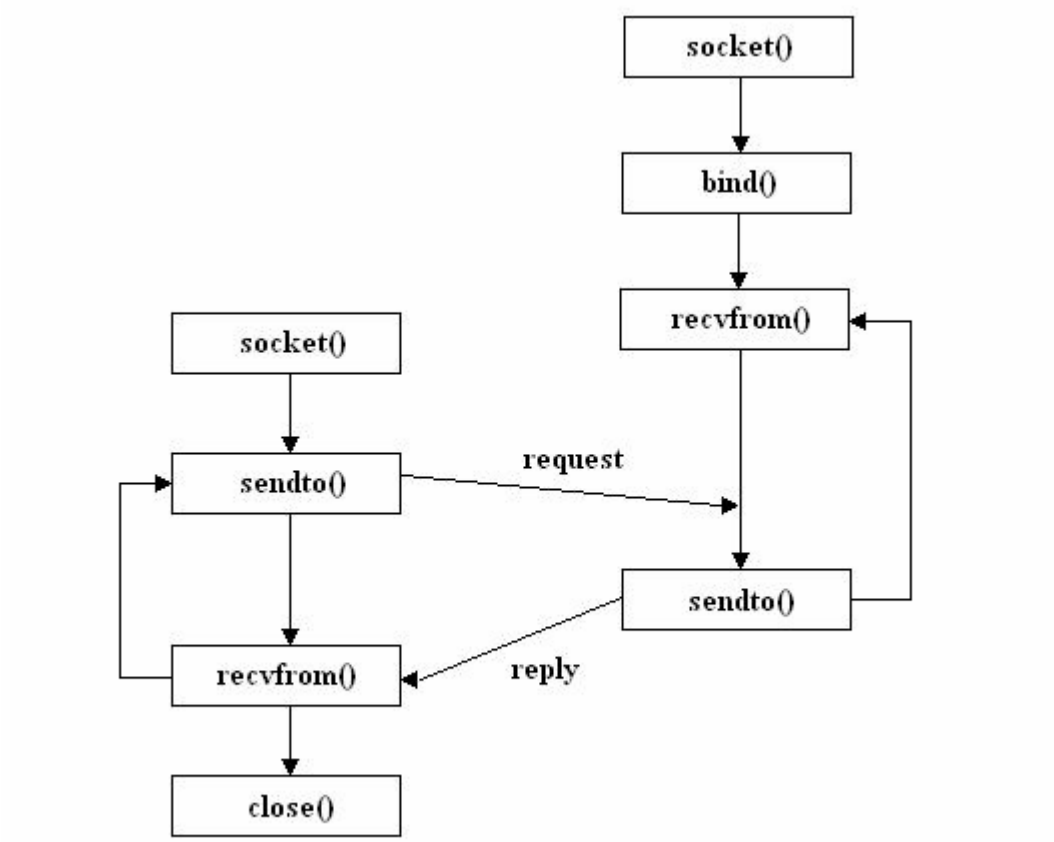
5.1 BSD sockety

Socket je koncovým bodom komunikácie a prvý krát sa objavil v operačnom systéme BSD. Ako analógiu socketu si môžeme predstaviť typický súbor, tzn. je to nejaký komunikačný nástroj, do ktorého môžeme dáta zapisovať, ale aj z neho čítať. Z nášho pohľadu je to teda niečo medzi naším aplikačným procesom napísaným v jazyku C a jadrom operačného systému, ktorý implementuje TCP alebo UDP protokol, na nižšej úrovni IP protokol.

Aplikácie vytvorené pomocou BSD socketov môžu využívať TCP alebo UDP protokol. Pretože multicast funguje nad protokolom UDP, ďalej sa už v práci budem zaoberať práve týmto protokolom. Problém, prečo nie je možné prenášať protokol TCP, je v spoľahlivosti. TCP vyžaduje potvrdzovanie posielaných dát, prípadne opätovné posielanie dát, ktoré sa po ceste stratili. Sieť by si musela pamätať, ktoré pakety už vyslala, a to by výrazne zaťažovalo routre. Ďalší problém sa vynára v spojitosti multicastu s TCP protokolom v tom, ako zabezpečiť, aby všetci záujemcovia zaručene dostali paket. Môže sa stať, že niektorí z príjemcov dáta dostanú, iní nie. Bolo by nutné udržiavať potrebné informácie, obsluhovať potvrdzovanie prijatých paketov a v prípade, že niektorí z nich dáta nedostali, zabezpečiť opakovaný prenos. Táto situácia by opäť výrazne zaťažovala, tak ako vysielaciu stranu, tak prenosové kapacity.

UDP je nespojovaný, nespoľahlivý protokol oproti protokolu TCP. Z toho taktiež vychádzajú vlastnosti multicastu, ktorý protokol UDP využíva.

Typická BSD socket aplikácia, teda aplikácia využívajúca protokoly TCP/IP je založená na modelu klient - server, čo znamená, že existujú aplikácie, ktoré prijímajú požiadavky od klientov a poskytujú služby (servery). Na druhej strane sú aplikácie (klienti), ktorí dané služby využívajú. Obrázok 14. ukazuje funkcie volané typickou UDP aplikáciou s modelom klient/server.



Obrázok 14: Príklad komunikácie UDP, vľavo UDP klient, vpravo UDP server. [20]

Na tomto obrázku funkcia *socket()* slúži na vytvorenie socketu a vracia deskriptor socketu. Funkcia *bind()* pomenuje socket, pomocou *connect()* sa pripojíme k serveru. Funkcie *recvfrom()* resp. *sendto()* príjmu resp. odošlú dáta prostredníctvom socketu. Funkcia *close()* ukončuje spojenie.

Ako je uvedené vyššie, pomocou socketov môžeme relatívne jednoducho vytvárať sieťové aplikácie. Existuje však socket, ktorý umožňuje so socketom pracovať na „nižšej“ úrovni. Nie však na nižšej úrovni OSI/ISO modelu, ale takej úrovni, kde môžeme priamo ovplyvňovať položky hlavičky paketu.

5.2 Vytvorené programy

Ako už bolo uvedené vyššie, programy, ktoré pri testovaní využívam sú naprogramované pomocou BSD socketov, v jazyku C. Pre implementáciu som zvolil operačný systém Linux. BSD sockety však poskytujú prácu aj v operačnom systéme Windows s minimálnou úpravou. Vďaka tomu je aplikáciu pre tento operačný systém možné upraviť jednoducho. Všetky programy sú vo forme zdrojových textov, preto je nutné ich pred použitím skompilovať. Kompilácia sa spustí príkazom make v príslušnom adresári daného programu. V nasledovných riadkoch priblížim prehľad týchto programov s popisom vysvetľujúcim implementáciu a zároveň aj funkciu daného programu.

5.2.1 SendMcast

Tento program je jedným z najzákladnejších programov, ktoré by nemali v tejto práci zaoberajúcej sa multicastom chýbať. Síce sa v ďalších programoch zaoberám touto problematikou iným spôsobom, než pomocou programu SendMcast, uvediem aspoň jeden úvodný test realizujúci overenie funkcie multicastu a konfigurácie routrov na multicastové routre. Taktiež pomocou programu SendMcast je možné vidieť silu multicastu v tom, ako stačí dáta jeden krát vyslať a všetci ostatní záujemcovia o dáta ich budú dostávať. Predpokladom však musí byť podpora zo sieťovej vrstvy. Týmto sa myslí predovšetkým podpora zo strany routrov. Úlohou programu je vyslanie UDP paketu na multicastovú skupinu.

Pre testovacie účely je program implementovaný tak, že vyslané dáta sú vopred dané. Taktiež z testovacích dôvodov je tento reťazec odoslaný viac krát na adresu a port zadanú ako argument.

5.2.2 GetMcast

Program GetMcast je podobný ako SendMcast. Využíva toho, že BSD sockety poskytujú dostatočne silné prostriedky pre prácu s multicastom. Oproti bežnej BSD aplikácii pre prenos dát je v programe uvedená jedna funkcia *setsockopt()* s parametrom IP_ADD_MEMBERSHIP. Táto funkcia obstará za programátora multicastovej aplikácie všetky dôležité úlohy. Základnou úlohou klienta, snažiaceho sa o príjem dát z nejakej multicastovej skupiny je predovšetkým prihlásenie sa k tejto skupine, čo táto funkcia zabezpečí. Avšak klient musí neustále odpovedať na dotazy routrov, či majú stále záujem byť prihlásený k multicastovej skupine a teda dostávať dáta z konkrétnej multicastovej skupiny. Programátor aplikácie musí toto zaobstarať. BSD sockety však túto úlohu v podobe funkcie *setsockopt()* vykonajú.

Program GetMcast je podobne ako SendMcast spúšťaný s parametrom. Prvým je multicastová skupina, druhým port.

5.2.3 SendIGMP

Predošlé programy využívali silu BSD socketov predovšetkým funkciou *setsockopt()*, kde stačilo okrem iného zadať parameter IP_ADD_MEMBERSHIP, či IP_DROP_MEMBERSHIP a daný program obstaral všetko potrebné. Vytvoril IGMP paket, ktorým sa program pripojil do požadovanej multicastovej skupiny. Na druhej strane daný mechanizmus neposkytuje až takú flexibilitu, ktorá je pri testovaní multicastu požadovaná. Popri inom nie je možné vytvoriť iné IGMP pakety, než sú funkciou *setsockopt()* poskytované. Vzhľadom k tomu som vytvoril tento program, ktorý umožňuje väčšie možnosti práce s multicastom a predovšetkým s protokolom IGMP.

Základom programu SendIGMP je štruktúra *igmp*, ktorou som si nadefinoval štruktúru IGMP paketu. Podľa obrázku 4. obsahuje IGMP paket *typ, checksum* a štruktúru *Group Address* obsahujúcu skupinu, ku ktorej sa daným IGMP paketom prihlasujem.

Prvým argumentom pri spúšťaní programu je typ IGMP správy podľa tabuľky 3, uvedenej v kapitole 6. Druhým parametrom je multicastová skupina, ku ktorej sa klient požaduje prihlásiť. Na základe tretieho argumentu, spolu s ktorým spúšťam program, určujem cieľ IGMP paketu, teda *Destination Address*. Pretože routre vysielajú IGMP pakety na multicastovú adresu, ku ktorej sa pripojujú, mal by byť tento paket smerovaný práve tam. Vo svojich testoch som sa pokúsil aj o iný prípad, kedy som vysielal paket napríklad na adresu 224.0.0.1, čo reprezentuje multicastovú adresu všetkých zariadení na danej sieti. Taktiež som otestoval vyslanie paketu s určitou multicastovou skupinou v tele paketu na ľubovolnú inú multicastovú adresu.

Socket je vytvorený ako RAW socket s parametrom IPPROTO_RAW, čím naznačujem, že som sa zaoberal tvorbou paketu na "nižšej" úrovni, teda vyplňoval jednotlivé položky IP paketu.

Jednotlivé položky IP paketu sú zaplnené nasledovne:

```
ip_header->ip_v  = 4;
ip_header->ip_hl  = 5;
ip_header->ip_tos  = 0;
ip_header->ip_id  = 28;
ip_header->ip_ttl  = 255;
ip_header->ip_p  = IPPROTO_IGMP;
ip_header->ip_sum  = 0;
ip_header->ip_dst.s_addr = s_addr_in.sin_addr.s_addr;
ip_header->ip_src.s_addr = 0;
igmp_header->igmp_type = igmp_typ;
igmp_header->igmp_code = 0;
igmp_header->igmp_cksum = 0;
inet_aton(skupina,&igmp_header->igmp_group);
```

Za zmienku stojí predovšetkým položka *igmp_type* kde sú zadávané hodnoty, na základe ktorých sú vytvárané správy protokolu IGMP. Pomocou funkcie *inet_aton()* zadávam do položky *igmp_group* multicastovú adresu, ku ktorej sa pripájam.

5.2.4 CaptureIGMP

Úlohou tohoto programu je zistiť, či na danej LAN sieti existuje multicast. Princíp programu spočíva v tom, že program po spustení načúva a snaží sa prijať ľubovolný IGMP paket. V prípade, že tento IGMP paket príjme, značí to, že na sieti je spustený multicast.

Sieťové rozhranie je bežne v stave, kedy prijíma len dáta určené pre ne samotné a zahadzuje všetky ostatné dáta. Zapnutie promiskuitného režimu však zabezpečí, že rozhranie bude prijímať aj také dáta, ktoré preň nie sú určené. Pôvodne som sa v tejto časti chcel zaoberať promiskuitným režimom pre príjem IGMP paketov. Týmto by som mohol program spustiť na ľubovolnom počítači v LAN sieti a tým zistiť, aké skupiny na danej sieti existujú. Pretože však sú IGMP pakety so správou

typu *Membership Query* doručované z routrov všetkým počítačom, ku ktorým sú pripojené rozhrania routra, nie je nutné sa promiskuitným režimom zaoberať.

Zo samotnej podstaty multicastu vyplýva nasledujúca nevýhoda programu CaptureIGMP. Multicastový router komunikuje protokolom IGMP s počítačmi, ktoré má pripojené ku svojím rozhraniam. Program spustený na jednom z týchto počítačov príjme spomínaný IGMP paket a určí, že je na sieti multicast. Problém sa vynára vo chvíli, keď je testovaná LAN sieť, na ktorej sú multicastové routre, ale aj routre, ktoré multicast nepodporujú. Spustením programu na počítači, ktorý je pripojený k rozhraniu routra bez podpory multicastu nie je možné zistiť, či je na danej LAN sieti multicast skutočne spustený. Nestačí preto prístup k ľubovolnému počítaču v sieti, ale je nevyhnutný prístup k počítaču, ktorý je pripojený k niektorému z multicastových routrov. Podobne, program dokáže zistiť, ktoré multicastové skupiny sú na danom multicastovom routry. Nedokáže však určiť ostatné multicastové skupiny v sieti, ktoré existujú na iných multicastových routroch. Test, ktorý súvisí s danou problematikou je uvedený v 6. kapitole, štvrtom teste.

V tomto programe sa opäť vyskytuje štruktúra *igmp*, ktorou reprezentujem IGMP paket. Socket je tentokrát vytvorený ako RAW socket s parametrom IPPROTO_IGMP, čím naznačujem, že sa jedná o pakety IGMP, resp. že program prijíma IGMP pakety.

V ďalších častiach programu po prijatí ľubovolného IGMP paketu program vypisuje jednotlivé položky IGMP paketu.

6 Návrh testov a ich realizácia

V tejto časti diplomovej práce uvádzam testy, ktoré som sa rozhodol realizovať. Súčasťou tejto kapitoly je zapojenie pri jednotlivých testoch, na ktorom je daná problematika z multicastu realizovaná. Ďalej priblížim programy, ktoré som vytvoril a pri testovaní použil, a taktiež problematiku, ktorú sa snažím daným testom preskúmať. Na záver každého testu zhrniem dosiahnuté výsledky a porovnám s očakávanými.

Pri rozhodovaní aké charaktery majú mať jednotlivý testy, ktoré realizujem, rozhodovali dva faktory. Na jednej strane sú to testy, ktoré ukazujú prirodzený význam multicastu. Tým myslím vyslanie dát a ich prijatie viacerými príjemcami, bez nutnosti vysielať dáta od zdroja viac krát. Pretože multicast ešte nie je bežne rozšírený, čitateľovi, ktorý sa s multicastom ešte prakticky nestretol môžu aj tieto testy viac danú problematiku priblížiť. Druhý typ testov realizujem s úmyslom testovať situácie, ktoré sa pravdepodobne v praxi ani často nevyskytujú, ale o to môže byť výsledok zaujímavejší. Príkladom je vyslanie IGMP paketu na adresu 224.0.0.1, alebo pripojenie počítača k dvom routrom paralelne a reakcia oboch zariadení pri prihlasovaní sa počítača k skupine.

Jednotlivé testy bolo možné realizovať na jednom univerzálnom zapojení. Zapojenie by mohlo pozostávať z viacerých počítačov, viacerých routrov a všetky testy by boli realizované na tomto zapojení. Je s tým spojená nevýhoda v podobe menšej prehľadnosti, či už skúmaných záležitostí, alebo dosiahnutých výsledkov. Preto som pre každý test použil iné zapojenie, ktoré je zjednodušené, ale dostačujúce pre daný problém, ktorý som testoval.

Súčasťou testov, resp. ich vyhodnocovania bola reakcia routrov. Inými slovami výpisy rôznych ladiacich záznamov udalostí.

Všetky konfigurácie k jednotlivým zapojeniam uvádzam v prílohách. Jedná sa o konfigurácie routrov, prípadne IP adresy a masky počítačov. Za zmienku stojí informácia o použitom unicastovom protokole, ktorý pre svoju činnosť potrebuje protokol PIM. Pri testoch som použil protokol EIGRP.

Pretože súčasťou testovania je konfigurácia a aj výpisy z routrov, uvádzam niekoľko príkazov použitých v ďalšej časti tejto kapitoly. Užívateľské rozhranie operačného systému používaného na routroch a switchoch firmy Cisco je delené do niekoľkých módov, pričom v každom móde umožňuje iné činnosti.

- **Užívateľský mód** – mód, v ktorom sa nachádzame hneď po prihlásení, má len obmedzené možnosti príkazov. Neumožňuje meniť konfiguráciu.

  ROUTER>

- **Privilegovaný mód** - tento mód umožňuje meniť konfiguráciu a zobrazovať rôzne údaje.

  ROUTER#

- **Konfiguračný mód** - poskytuje konfiguračné funkcie.

  ROUTER(config)#

- **Konfigurácia rozhraní** - konfigurácia vlastností určitého rozhrania.

ROUTER(config-if)#

Zoznam použitých príkazov:

Router(config)# **ip multicast-routing** - povoľuje IP multicast smerovanie a zobrazovanie informácií o multicastových skupinách,

Router(config-if)# **ip pim dense-mode** - povoľuje PIM protokol na danom rozhraní s módom dense-mode,

Router(config)# **ip pim sparse-mode** - povoľuje PIM protokol s módom sparse-mode,

Router# **debug ip igmp** - zobrazuje prijaté a vyslané IGMP pakety,

Router# **debug ip packet** - zobrazuje prijaté a vyslané IP pakety,

SW1# **ip igmp snooping** – zapína IGMP snooping.

Ďalšie príkazy budú zobrazovať informácie:

Router# **show ip igmp groups** - zobrazuje multicastové skupiny, priamo pripojené k routru,

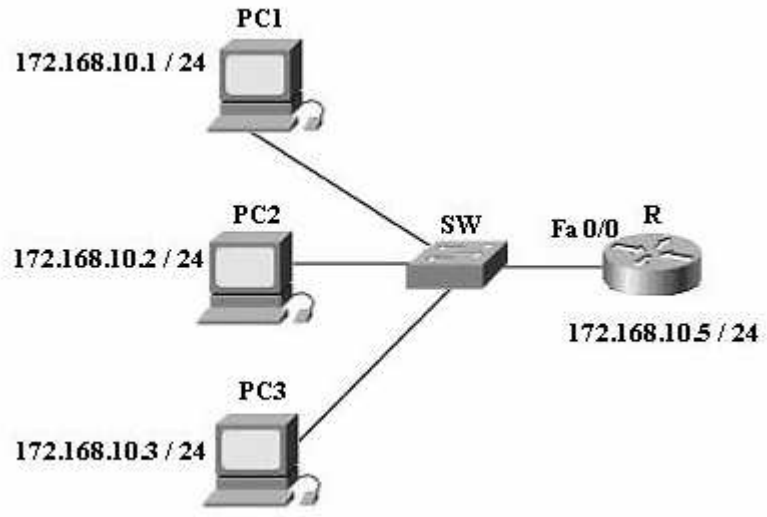
Router# **show ip mroute** - ukazuje obsah multicastovej smerovacej tabuľky,

SW1# **show mac address-table multicast** – zobrazuje záznamy zo smerovacej tabuľky switcha.

6.1 Test č. 1

Prvé testované zapojenie predpokladá jeden router, jeden switch, prostredníctvom ktorého budú pripojené počítače.

Zapojenie č.1 je na obrázku 15.



Obrázok 15: Prvé testované zapojenie.

Použité programy:

- SendIGMP
- CaptureIGMP
- GetMcast

Priebeh testovania:

Po úvodnej konfigurácii zapojenia, ktorú uvádzam v prílohe 1, nie je ešte multicast žiadnym spôsobom podporovaný. Spusteným programom SendIGMP na jednom z počítačov vyšlem IGMP paket na ľubovolnú multicastovú skupinu. Pretože však na routry nie je spustený príkaz, ktorý by umožňoval akýmkoľvek spôsobom pracovať s multicastom, daný IGMP paket bude na routry zahodený.

Po úvodnom nakonfigurovaní routra a spustením nasledujúcich príkazov nastáva povolenie k multicastovému smerovaniu na routry.

R(config)# **ip multicast routing**

R(config-if)# **ip pim dense-mode**

Akonáhle je povolený PIM - DM na rozhraní routra, začne tento router vysielať IGMP paket na adresu 224.0.0.13 reprezentujúci *PIM Hello* paket a 224.0.1.40 ako multicastová skupina pre všetky PIM routre.

Adresa 224.0.0.13 nie je v tejto chvíli zaujímavá, pretože router sa snaží nájsť susedov, tzn. routre, na ktorých je tiež spustený PIM - DM. V tomto zapojení je k dispozícii len jeden router, žiadnu odpoveď router neprijme. Podobne je to s adresou 224.0.1.40, kde sa router automaticky pripojí do tejto multicastovej skupiny ako do skupiny, ku ktorej sa pripájajú všetky PIM routre. Keďže opäť je v zapojení len jeden router, iný okrem už zapojeného sa do danej skupiny neprihlási.

Ukážka ladiaceho výstupu z routra:

```
*Apr 9 17:05:05.555: IP: s=172.168.10.5 (local), d=224.0.0.1 (FastEthernet0/0), len 28, sending broad/multicast
*Apr 9 17:05:05.555: IGMP(0): WAVL Insert group: 224.0.1.40 interface: FastEthernet0/0Successful
*Apr 9 17:05:05.555: IGMP(0): Send v2 Report for 224.0.1.40 on FastEthernet0/0
*Apr 9 17:05:05.555: IGMP(0): Received v2 Report on FastEthernet0/0 from 172.168.10.5 for 224.0.1.40
*Apr 9 17:05:05.555: IGMP(0): Received Group record for group 224.0.1.40, mode 2 from 172.168.10.5 for 0 sources
```

Uvedený výpis znázorňuje postupnú situáciu, kedy z routra R zobrazeného v obrázku 15. s adresou 172.168.10.5, vysiela dotaz na adresu 224.0.0.1, teda na všetky zariadenia v sieti. Na tento IGMP paket samotný router odpovedá prihlásením svojho rozhrania FastEthernet0/0 do spomínanej skupiny 224.0.1.40.

Pomocou príkazov *show ip igmp groups* je vidieť, ako si router vytvára údaje o tom, v ktorých skupinách je prihlásený. V tejto chvíli je však možné spustiť programy, SendIGMP a CaptureIGMP.

Pomocou SendIGMP je možné vysielať pakety IGMP s rôznym typom správy podľa tabuľky 3.

Typ IGMP paketu	Názov
0x11	IGMP Membership Query
0x12	IGMP v1 Membership Report
0x16	IGMP v2 Membership Report
0x17	IGMP v2 Leave Group

Tabuľka 3: Typy správ IGMP protokolu.

Napríklad hodnota 0x12 znamená IGMP v1 *Membership Report*, teda IGMP protokol verzie 1 a typ *Membership Report*, znamenajúci požiadavok na členstvo v skupine. Hodnotu 0x16 reprezentuje IGMP protokol verzie 2 s typom správy *Membership Report*. Touto správou dáva stanica routru informáciu o tom, že chce dáta prijímať a prihlasuje sa do skupiny.

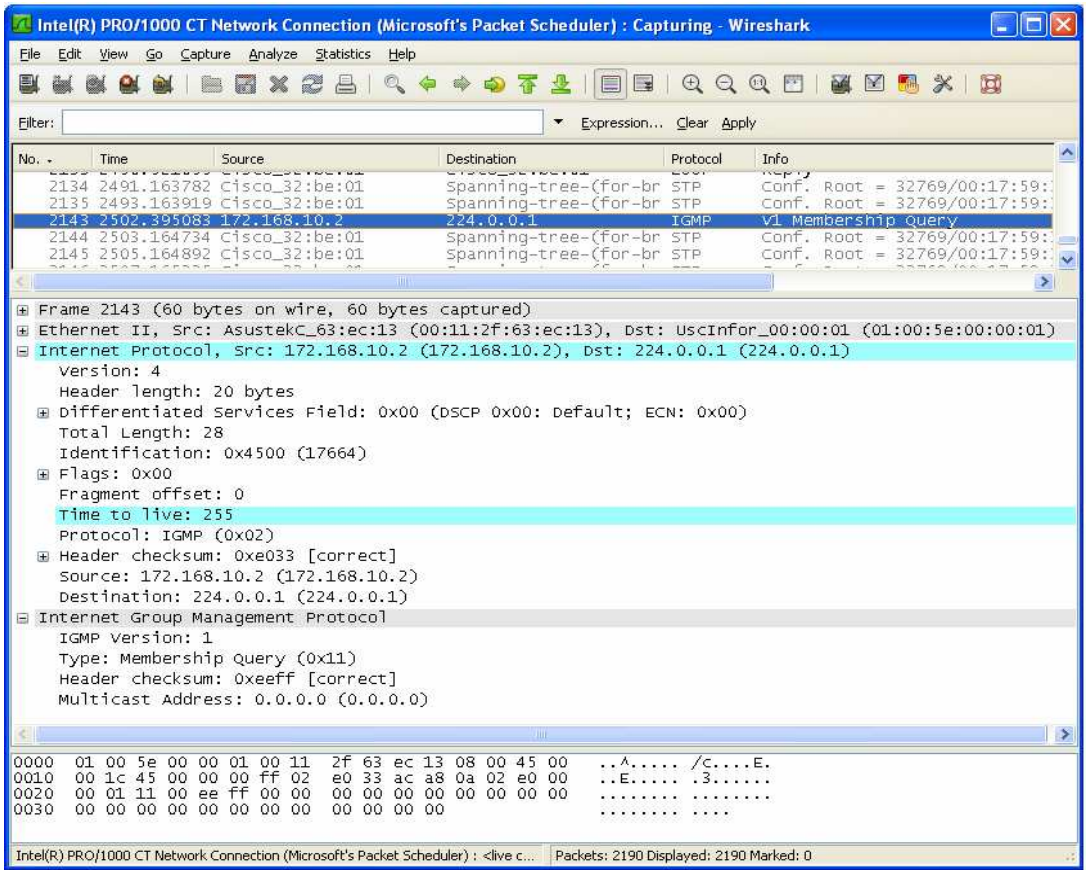
Týmto spôsobom je možné zadávať ďalšie hodnoty v položke typu IGMP paketu a vytvárať tak ďalšie protokoly využívajúce protokol IGMP. V tejto súvislosti spomeniem:

- protokol PIM s *Register* správou alebo *Join/Prune* správou,
- DVMRP protokol so správou *Graft*, *Prune*, okrem iných.

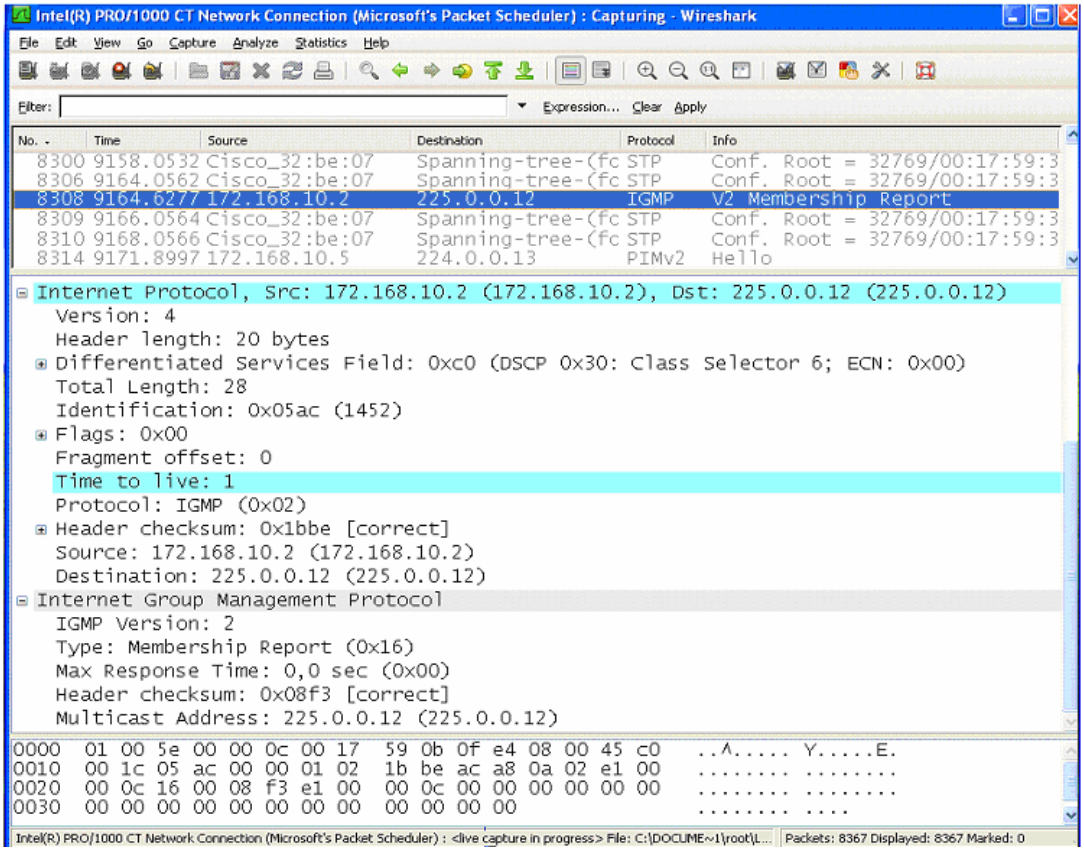
Tieto protokoly sú však už nad rámec mojej diplomovej práce, keďže primárne sa zaoberám protokolom IGMP a protokolom PIM.

V nasledujúcich riadkoch priblížim program Ethereal, ktorý som mal spustený na počítači PC3. Tento program umožňuje zachytávať pakety prenášané sieťou. Týmto som chcel ozrejmiť štruktúru a obsah konkrétneho paketu protokolu IGMP, keďže Ethereal ponúka prehľadné zobrazenie protokolov.

Na obrázku 16. a 17. sú prezentované spomínané IGMP pakety verzie 1 a verzie 2, vyslané programom, ktorý som vytvoril - SendIGMP a zachytené programom Ethereal.

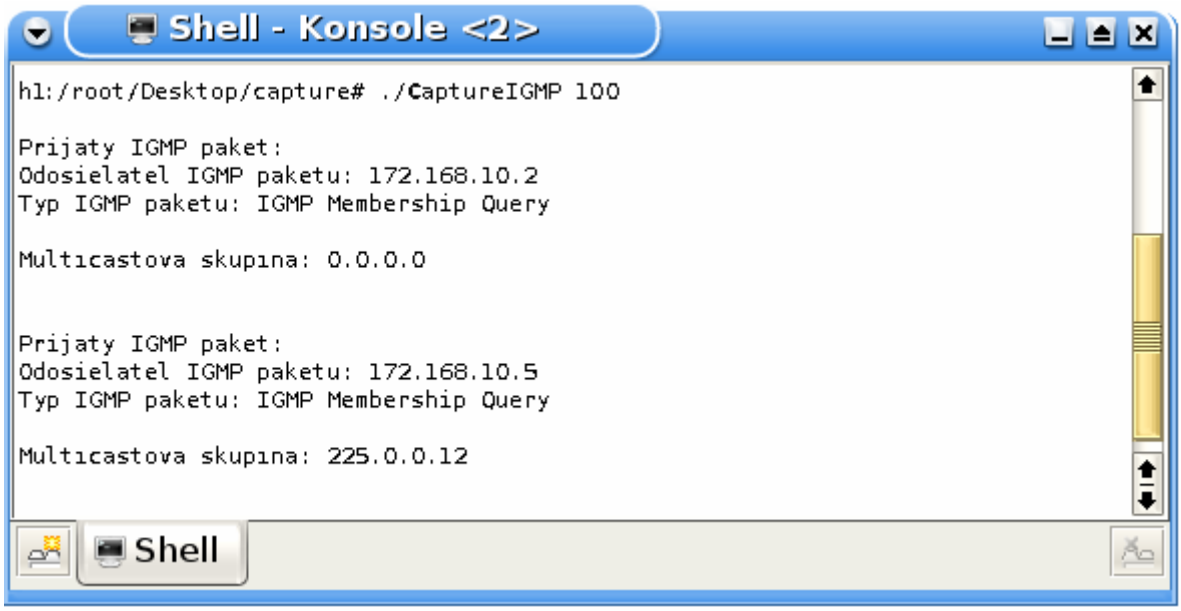


Obrázok 16: Protokol IGMP v1 *Membership Query*.



Obrázok 17: Protokol IGMP v2 *Membership Report*.

Podobne ako program Ethereal zachytí okrem iných aj tieto pakety, program CaptureIGMP dokáže zachytiť IGMP pakety. Program taktiež rozlíši jednotlivé typy správ tohoto protokolu. Ukážka je na obrázku 18.



V tomto okamžiku je pomocou programu SendIGMP odoslaný IGMP paket z počítača PC2 podľa obrázku 15, ktorého adresa je 172.168.10.2. Tento paket je vyslaný s požiadavkou na prihlásenie k mutlicastovej skupine 225.0.0.12.

```
*Apr  9 17:08:55.267: IP: s=172.168.10.2 (FastEthernet0/0), d=225.0.0.12,
len 28, rcvd 2
*Apr  9 17:08:55.267: IGMP(0): Received v2 Report on FastEthernet0/0 from
172.168.10.2 for 225.0.0.12
*Apr  9 17:08:55.267: IGMP(0): Received Group record for group 225.0.0.12,
mode 2 from 172.168.10.2 for 0 sources
*Apr  9 17:08:55.267: IGMP(0): WAVL Insert group: 225.0.0.12 interface:
FastEthernet0/0Successful
```

Vyššie zobrazený výpis vypovedá o vyslaní paketu z adresy 172.168.10.2 na multicastovú adresu 225.0.0.12. Router rozpozná, že sa jedná o IGMP paket a prihlási svoje rozhranie FastEthernet0/0 k multicastovej skupine 225.0.0.12.

Následne sú priblížené výpisy routra po zadaní príkazov, ktoré bližšie ukazujú, že router na rozhraní FastEthernet0/0 je prihlásený ku skupine 224.0.1.40 a už aj ku skupine 225.0.0.12.

R# **show ip igmp groups**

```
IGMP Connected Group Membership
Group Address    Interface                Uptime    Expires   Last
225.0.0.12       FastEthernet0/0          00:00:14  00:02:45  172.168.10.2
224.0.1.40       FastEthernet0/0          00:04:04  00:02:02  172.168.10.5
```

V stĺpci *Group Address* sú uvedené spomenuté adresy, ku ktorým je router aj počítač PC2 na rozhraní FastEthernet0/0 prihlásený.

R# **show ip membership**

```
 Channel/Group      Reporter        Uptime   Exp.    Flags    Interface
 *,225.0.0.12       172.168.10.2    00:00:36 02:23   2A       Fa0/0
 *,224.0.1.40       172.168.10.5    00:04:26 02:43   2LA      Fa0/0
```

Tento príkaz zobrazil skupiny, ku ktorým je router pripojený.

R# **show ip mroute**

```
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
(*, 225.0.0.12), 00:08:12/00:01:29, RP 0.0.0.0, flags: DC
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    FastEthernet0/0, Forward/Dense, 00:04:20/00:00:00
(*, 224.0.1.40), 00:37:32/00:02:57, RP 0.0.0.0, flags: DCL
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    FastEthernet0/0, Forward/Dense, 00:37:32/00:00:00
```

Zadaný príkaz zobrazuje multicastovú smerovaciu tabuľku, kde sú informácie o požadovanej multicastovej skupine 225.0.0.12, a zároveň multicastová skupina 224.0.1.40 pre všetky PIM routre.

Pretože router sa periodicky dotazuje na členstvo v skupine, je nutné neustále odpovedať na dotaz routru, inak bude počítač zo skupiny odstránený. Taktiež je odhlásenie možné uskutočniť explicitne. Po vyslaní IGMP paketu so správou typu *Leave Group* z programu SendIGMP je daná skupina odhlásená.

```
*Apr  9 17:27:41.683: IP: s=172.168.10.2 (FastEthernet0/0), d=224.0.0.2,
len 28, rcvd 0
*Apr  9 17:27:41.683: IGMP(0): Received Leave from 172.168.10.2
(FastEthernet0/0) for 225.0.0.12
*Apr  9 17:27:41.683: IGMP(0): Received Group record for group 225.0.0.12,
mode 3 from 172.168.10.2 for 0 sources
*Apr  9 17:27:41.683: IGMP(0): Lower expiration timer to 2000 msec for
225.0.0.16 on FastEthernet0/0
```

```
*Apr  9 17:27:41.687: IGMP(0): Send v2 Query on FastEthernet0/0 for group
225.0.0.16
```

Z uvedeného je možné si všimnúť zaujímavú skutočnosť. Router príjme *Leave Group* správu, multicastovú skupinu teda odhlasuje, ale ihneď nato odosiela dotaz na túto skupinu, či ešte niektorý záujemca na tomto rozhraní nezostal s požiadavkou na príjem z tejto multicastovej skupiny. V tomto prípade na dotaz router odpoveď nedostane, keďže už žiadny počítač záujem neprejaví. Uvedená multicastová skupina je z príslušného rozhrania na routry odhlásená.

Teraz popíšem funkciu *setsockopt()* z BSD socketov, ktorá sa postará o všetky nevyhnutné procesy, ktoré musia byť uskutočnené. Použitý je k tomu program GetMcast, pomocou ktorého sa pripojí klient z počítača k multicastovej skupine. Parametrom IP_ADD_MEMBERSHIP funkcia zabezpečí posielanie IGMP paketu. Router sa klienta neustále dotazuje na pretrvávajúci záujem o členstvo v skupine. Táto funkcia zaistí aj to, že na každý dotaz routra bude pravidelne odpovedané a nedôjde tým k odhláseniu zo skupiny. V tomto stave je program pripravený prijímať dáta z multicastovej skupiny. Pretože však toto zapojenie nie je ideálne pre testovanie funkčnosti multicastu, túto funkčnosť som overil v ďalšom zapojení.

IGMP paket je štandardne posielaný na adresu danej multicastovej skupiny. Prislúchajúci router tento paket zachytí a obslúži. Zaujímavú možnosť však ponúka situácia, kedy paket vyšleme na adresu 224.0.0.1 reprezentujúcu všetky multicastové zariadenia na sieti. Taktiež táto situácia vyžaduje iné zapojenie, preto ju uvediem neskôr.

Nakoľko som v úvode uviedol názorné ukážky zachytených IGMP paketov programom Ethereal, otestoval som nasledujúcu skutočnosť. Router, ktorý príjme IGMP verzie 1 detekuje, že sa jedná o nižšiu verziu a daný paket spracuje. Z programu SendIGMP posielam IGMP paket správy *Membership Report* verzie 1 s pokusom o prihlásenie ku skupine 226.0.0.36.

```
*Apr  9 17:13:53.399: IP: s=172.168.10.2 (FastEthernet0/0), d=226.0.0.36,
len 28, rcvd 2
*Apr  9 17:13:53.399: IGMP(0): Received v1 Report on FastEthernet0/0 from
172.168.10.2 for 226.0.0.36
*Apr  9 17:13:53.399: IGMP(0): Received Group record for group 226.0.0.36,
mode 2 from 172.168.10.2 for 0 sources
*Apr  9 17:13:53.399: IGMP(0): WAVL Insert group: 226.0.0.36 interface:
FastEthernet0/0Successful
```

Daný výpis ukazuje prihlásenie klienta do skupiny 226.0.0.36 protokolom IGMP v1. Čo sa ale stane v prípade, keď sa ku skupine klient prihlási jednou verziou IGMP protokolu, a pokúsi sa odhlásiť druhou? Skúsil som sa preto v situácii, kedy je prihlásený ku skupine protokolom verzie 1, odhlásiť IGMP protokolom verzie 2. Router daný IGMP paket so správou *Leave Group* v poriadku príjme, o čom vypovedajú nasledujúce riadky. Router prijal paket z adresy 172.168.10.2 týkajúci sa multicastovej skupiny 226.0.0.36.

```
*Apr  9 17:21:36.719: IP: s=172.168.10.2 (FastEthernet0/0), d=224.0.0.2,
len 28, rcvd 0
*Apr  9 17:21:36.723: IGMP(0): Received Leave from 172.168.10.2
(FastEthernet0/0) for 226.0.0.36
*Apr  9 17:21:36.723: IGMP(0): Received Group record for group 226.0.0.36,
mode 3 from 172.168.10.2 for 0 sources
```

Toto odhlásenie sa ale nepodarí, pretože po zadaní príkazu **sh ip igmp groups** stále zostáva dané rozhranie routra prihlásené k skupine. Po pokuse odhlásenia z tejto skupiny správnou verziou je naopak úspešne odhlásený. Tak ako som už uvádzal v teoretickej časti tejto práce, IGMP protokol verzie 1 neobsahuje správu *Leave Group*. V prípade, že je klient zaregistrovaný k multicastovej skupine IGMP v1, a dostane IGMP paket správy *Leave Group*, je táto správa na routry ignorovaná. Ďalšia možnosť ako otestovať to, či je možné záujemcu o multicastové dáta prihlásiť jednou verziu IGMP a druhou odhlásiť, sa vynára s použitím IGMP v3 protokolu. Preto sa pokúsim vyslať protokol IGMP v3 so správou *Membership Report*. Router danú správu príjme.

```
*Apr  10 14:00:51.919: IP: s=172.168.100.2 (FastEthernet0/1),
d=225.0.0.22, len 28, rcvd 2
*Apr  10 14:00:51.923: IGMP(0): v3 report on interface configured for
v2, ignored
```

Daný paket je ale na routry ignorovaný. Usudzujem, že IGMP v3 nie je na danom routry podporovaný. Z tohoto dôvodu sa mi nepodarilo zistiť, či je možné klienta prihlasovať a odhlasovať rôznymi verziami protokolu IGMP.

Na záver tohoto testu ešte uvediem jeden postreh, ktorý som mi pri testovaní zistil. Prihlasovaním sa do skupiny z počítača je vysielaný IGMP paket so správou *Membership Report*, ktorý v tele obsahuje multicastovú skupinu, na ktorú má záujem sa prihlásiť. Celý paket má ako cieľovú adresu uvedenú tú istú požadovanú multicastovú skupinu. Pri testovaní sa mi podarilo vyslať paket aj s rôznou multicastovou skupinou. V tele IGMP paketu je požadovaná multicastová skupina ale rozdielna s cieľovou skupinou. Zistil som, že táto skutočnosť vôbec neovplyvní ďalšie správanie routra. Router jednoducho príjme paket, prečíta hodnotu položky paketu **igmp_group** a na základe adresy prihlási klienta k multicastovej skupine.

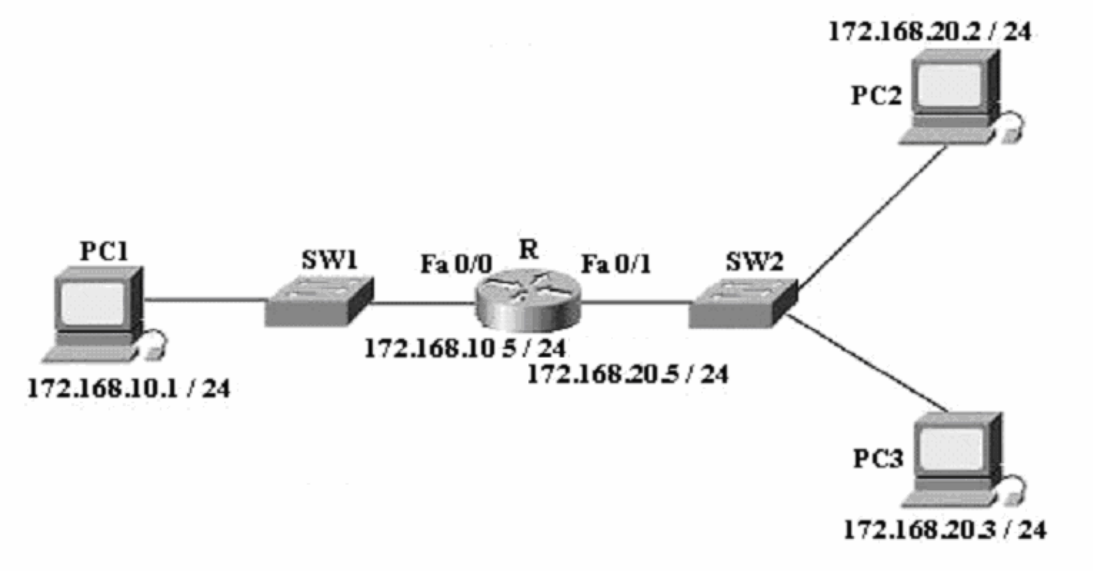
Zhrnutie:

V uvedenom zapojení bolo testované klasické zapojenie obsahujúce router a počítače, ktoré sú k nemu pripojené. V teste som ukázal činnosť routra okamžite po prihlásení, kedy sa snaží prihlásiť do multicastovej skupiny pre všetky routre. Ďalej som uviedol funkciu programu SendIGMP. Tento program má prínos predovšetkým v možnosti flexibilne pracovať s protokolom IGMP, prihlasovať, resp. odhlasovať klientov z multicastových skupín. Taktiež je možná aj iná než typická práca s protokolom IGMP. Ako príklad, ktorý som použil uvediem prácu s rôznymi verziami IGMP protokolu.

6.2 Test č. 2

Toto zapojenie obsahuje jeden router, dva switche pre pripojenie k routru a dva počítače, paralelne pripojené k switchu SW2.

Zapojenie č. 2 je na obrázku 19.



Obrázok 19: Druhé testované zapojenie.

Cieľ testovania:

Porovnanie prostriedkov BSD socketov s „ručným“ prihlasovaním pomocou IGMP, IGMP snooping.

Použité programy:

- SendMcast
- GetMcast
- CaptureIGMP
- SendIGMP

Priebeh testovania:

V zapojení sa snažím poukázať predovšetkým na všeobecný význam multicastu, hlavne nato, ako stačí dáta jeden krát vyslať. Úlohou záujemcov o tieto dáta je len prihlásiť sa do skupiny. O všetko ostatné sa starajú multicastové routre, ktoré zabezpečia, aby jeden krát vyslané dáta boli doručené viacerým príjemcom. Taktiež porovnanie prostriedkov BSD socketov pre prácu s multicastom, a prácou s multicastom na nižšej úrovni. Druhou testovanou problematikou je technika IGMP snooping, ako výrazne môže ovplyvňovať zahlcovanie prenosových ciest.

Na počítači PC1 je spustený program SendMcast, ktorý vysiela multicastové dáta. Tento program využíva možnosti BSD socketov pre prácu s multicastom. Na počítačoch PC2 a PC3 je spustený program GetMcast, taktiež využívajúci multicastových schopností BSD socketov. Program GetMcast pracuje nasledovne. Pomocou funkcie *setsockopt()* s parametrom

IP_ADD_MEMBERSHIP uskutoční všetko potrebné pre prihlásenie do skupiny. Zabezpečí vyslanie IGMP paketu z príslušného počítača, na ktorom je program spustený a prihlási takto klienta do skupiny. Pretože prihlásený klient je k multicastovému routru prihlásený len určitú dobu, po uplynutí ktorej bude zo skupiny odhlásený, program musí umožňovať udržiavať klienta stále prihláseného. Toto zabezpečujú BSD sockety tým, že na dotaz routra, na IGMP paket so správou *Membership Query* odpovedajú IGMP paketom s úmyslom stále zotrvať prihlásený k danej skupine. Pre užívateľa je to veľmi výhodné, nakoľko stačí spustiť funkciu *setsockopt()* a o všetko ostatné je postarané. Tak ako v predošlom teste, tak aj v nasledujúcich sa prevažne zaoberám tvorbou a vysielaním samotného IGMP paketu, ktorý mi za účelom testovania multicastu prišiel vhodnejší a flexibilnejší.

Druhú záležitosť, ktorú sa snažím testovať v tomto zapojení je IGMP snooping. Táto vlastnosť switcha umožňuje nahliadať do multicastových dát a rozhodovať sa na ich základe, kam dáta vyslať a kam nie. V zapojení je ako prvá testovaná situácia, kedy je IGMP snooping povolený. Vyslaním dát na multicastovú skupinu, ku ktorej sú počítače PC2 a PC3 prihlásené dostávajú títo klienti všetok multicastový traffic vysielaný na túto skupinu. Pretože switch s nezapnutým IGMP snoopingom vysiela dáta na všetky svoje rozhrania, v tejto konkrétnej situácii IGMP snooping nezastáva žiadnu úlohu. V prípade, že IGMP snooping vypnem, budú počítače dostávať dáta podobne ako so zapnutým IGMP snoopingom. Keď sa užívateľ na počítači PC2 rozhodne, že už nemá o multicastové dáta záujem, ukončí program GetMcast. Switch dokáže zistiť, že klient na niektorom z rozhraní switcha nemá o dáta záujem, nebude mu už multicastový traffic preposielať. Skutočný prínos IGMP snoopingu možno vidieť až v nasledujúcej situácii. Na switchi vypnem IGMP snooping príkazom:

SW1# **no ip igmp snooping**

V tomto okamžiku začne switch všetok multicastový traffic, ktorý dostane na niektorom zo svojich rozhraní preposielať na všetky ostatné rozhrania. Bez ohľadu nato, či má klient pripojený k tomuto rozhraniu o dáta záujem. Túto situáciu som otestoval s použitím programu GetMcast. Tento program prijíma pakety vyslané na multicastovú skupinu. Z počítača PC1 je programom SendMcast vyslaný testovací reťazec. Obidva počítače, PC2 aj PC3 dáta dostanú, aj napriek tomu, že klient na PC2 o tieto dáta už nemá záujem. V tomto prípade je možné vidieť, ako výrazne je možné ušetriť prenosových kapacít použitím IGMP snoopingu. Môže nastať situácia, že z počítača PC1 bude multicastom vysielané video. Už aj v takomto jednoduchom zapojení, ktoré pozostáva z dvoch príjemcov, je použitím IGMP snoopingu vidieť výrazne ušetrenie prenosových kapacít.

V ďalšej časti testu, poukážem na funkciu programu, pre pasívny aj aktívny listening k zisťovaniu, či je v danej LAN sieti multicast. Druhé testované zapojenie je vhodné k úvodnému vysvetleniu funkcie aktívneho a pasívneho listneningu. Po pochopení princípu multicastu, ktorý je súčasťou teoretickej časti diplomovej práce je situácia výrazne zjednodušená. Router, ktorý má zapnutý multicast a napríklad protokol PIM s režimom Dense Mode, vysiela v približne 2 minútovom intervale dotaz na členstvo v skupine 224.0.1.40. Táto skupina reprezentuje všetky PIM routre na sieti. Aj v prípade, že je v zapojení len jeden router, je prihlásený do tejto skupiny a neustále sa na

svojich rozhraniach dotazuje za účelom zistenia ďalších PIM routrov. Danú situáciu možno využiť práve v pasívnom listeningu, kedy stačí zapnúť program CaptureIGMP a čakať práve na zmienený dotaz na multicastovú skupinu.

Podobne funguje aktívny listening. Vyšlem dotaz práve na skupinu 224.0.1.40. Oklameme router, pretože mu oznámim, že na sieti, presnejšie na rozhraní, z ktorého sa pripájam, je ďalší PIM router. Router na daný dotaz odpovedá IGMP paketom smerovaným na adresu 224.0.1.40. Prijatím tohoto IGMP paketu je jasné, že na sieti je multicast. Zložitejšia situácia vznikne v prípade, ak je na sieti viac zapojených routrov. Túto situáciu prezentujem štvrtým testovaným zapojením.

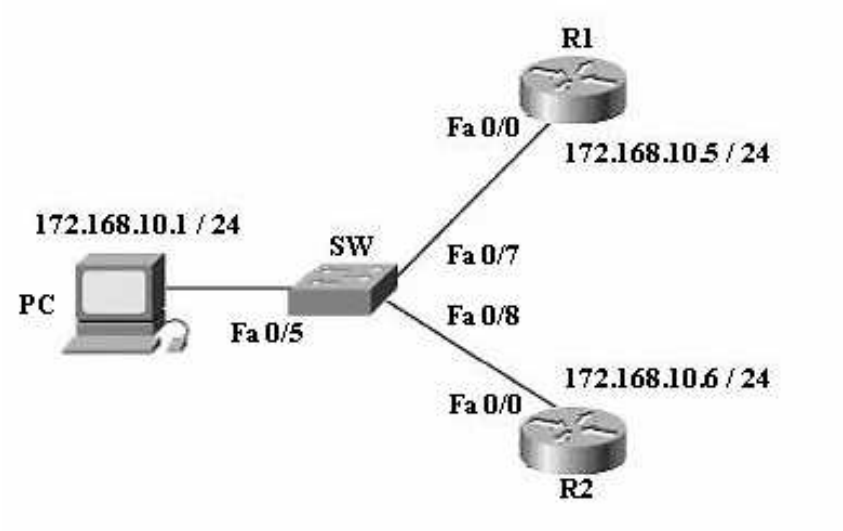
Zhrnutie:

Prezentované zapojenie je ideálne pre poukázanie na všeobecný prínos multicastu. Jeden vysielateľ dát, jedno vyslanie, viacerí príjemcovia. O všetko potrebné sa postaral multicastový router. S uvedeným úzko súvisí IGMP snooping. Táto vlastnosť switcha umožňuje nahliadať do multicastových dát a smerovať ich podobne ako router. Bez podpory IGMP snoopingu na switchi nastáva výrazne nadmerné zaťažovanie prenosových kapacít, ktoré vidieť už v takomto jednoduchom zapojení. Ukázané sú aj programy SendMcast a GetMcast, ktorých výhody plynú z výhod multicastu. Sú implementované pomocou BSD socketov, ktoré výraznou mierou podporujú multicast.

6.3 Test č. 3

Toto zapojenie obsahuje dva routre, jeden switch, prostredníctvom ktorého budú tieto routre paralelne pripojené k počítaču.

Zapojenie č.3 je na obrázku 20.



Obrázok 20: Tretie testované zapojenie.

Cieľ testovania:

Prihlasovanie klienta do skupiny, IGMP snooping, mapovanie multicastových skupín.

Použité programy:

- SendIGMP

Priebeh testovania:

Zapojenie, ktoré tu prezentujem možno považovať za istým spôsobom umelo vytvorené, ale v situácii, kedy som použil záložný router k pôvodnému, získal opodstatnenie. V tomto zapojení testujem, ako sa zachovajú routre v prípade, že sa chce počítač pripojiť do určitej multicastovej skupiny a zároveň, ako túto situáciu ovplyvňuje IGMP snooping. Zapnutý IGMP snooping dovoľuje switchu skúmať získané multicastové dáta a na základe zákonitostí popísaných vyššie sa môže rozhodnúť, na ktoré z rozhraní paket prepošle, resp. neprepošle. Ako to ale bude s IGMP paketom, ktorým sa počítač snaží pripojiť do skupiny?

Počítač pripojený k switchu má štandardne nastavenú bránu, čo znamená adresu rozhrania routra, kam má prednastavene vysielať dáta. S IGMP paketom sa však pracuje inak. Túto sitáciu som otestoval nasledovne.

Po nakonfigurovaní zapojenia, upozorňujem nato, že IGMP snooping je automaticky zapnutý a preto ho v prvom kroku vypnem.

SW1# **no ip igmp snooping**

V tejto situácii by sa dalo očakávať, že sa daný počítač vyslaním IGMP paketu zrejme pripojí k obidvom routrom, teda, že si oba routre pridajú túto skupinu do multicastových smerovacích tabuliek. V tejto chvíli vyšlem IGMP paket s úmyslom prihlásiť sa na multicastovú skupinu 227.0.0.1 pomocou SendIGMP.

Ukážka výsledku pomocou príslušného príkazu.

R1# **show ip igmp groups**

```
IGMP Connected Group Membership
Group Address    Interface                Uptime    Expires   Last
227.0.0.1        FastEthernet0/0          00:00:07  00:02:52  172.168.10.1
224.0.1.40       FastEthernet0/0          00:08:03  00:02:17  172.168.10.5
```

R2# **show ip igmp groups**

```
IGMP Connected Group Membership
Group Address    Interface                Uptime    Expires   Last
227.0.0.1        FastEthernet0/0          00:00:01  00:02:58  172.168.10.1
224.0.1.40       FastEthernet0/0          00:07:40  00:02:23  172.168.10.5
```

Ako je vidieť, podľa očakávania nastáva situácia, že daný počítač je prihlásený k obom routrom. A ako túto situáciu ovplyvní IGMP snooping?

Zapnem IGMP snooping príkazom:

SW1# **ip igmp snooping**

Po vykonaní príkazu opäť vyšlem IGMP paket programom SendIGMP. Výsledok na oboch routroch je nasledovný. Po zadaní príkazu **show ip igmp groups** na obidvoch routroch R1 a R2 bude výsledok rovnaký s vyššie uvedenými.

IGMP snooping pracuje tak, že umožňuje switchu "nahliadnuť" do multicastových dát. Switch si do svojej pamäte ukladá príjemcov multicastového trafficu na základe odpovedí, na správu typu *Membership Report*. V tomto prípade však nemá v pamäti z predchádzajúcej činnosti žiadne informácie, nevie teda, kam by mal dáta preposlať. Z toho dôvodu ich prepošle na všetky zo svojich rozhraní. Vzhľadom k tomu IGMP snooping nijak neovplyvňuje prihlasovanie klientov do multicastových skupín. V zapojení prezentovaného charakteru bude klient prihlásený aj k viacerým routrom.

Ďalšia vlastnosť multicastu vo všeobecnosti je mapovanie multicastových skupín. Tak ako som už uviedol v teoretickej časti diplomovej práce, pri mapovaní multicastovej IP adresy na MAC adresu dochádza k určitej strate informácií, kedy sa 5 bitov z IP adresy nezúčastňuje tohoto mapovania. Inými slovami dochádza k tomu, že za určitých okolností môžu byť dve (dokonca až 2^5) rôzne multicastové IP adresy namapované na rovnakú MAC adresu.

V nadväznosti na predchádzajúci príklad uvediem tabuľku switcha, kde má uložené informácie z jeho vlastnej smerovacej tabuľky. Klient je prihlásený ku skupine 227.0.0.1.

SW1# **show mac address-table**

```
Vlan    Mac Address       Type        Ports
----    -----------       ----        -----
   1    0100.5e00.0001    IGMP        Fa0/5, Fa0/7, Fa0/8
   1    0100.5e00.0128    IGMP        Fa0/7, Fa0/8
```

Podľa zapojenia 3, z obrázku 20. vyplýva, že do skupiny 224.0.1.40 sú pripojené zariadenia na rozhraní Fa0/7 a Fa0/8, čo reprezentujú oba routre. Do skupiny 227.0.0.1 sú pripojené všetky tri zariadenia, teda obidva routre a klient PC1 na adrese 172.168.10.1. Uvedené MAC adresy sú zobrazené v hexa tvare, ktoré po prevedení do dekadickej sústavy reprezentujú príslušné multicastové adresy. V tejto chvíli programom SendIGMP posielam paket na adresu 228.0.0.1 s úmyslom prihlásiť sa. Práve táto adresa po prevední na MAC adresu bude zhodná s MAC adresou v pôvodnom IP tvare 227.0.0.1. Switch bude tieto dve adresy považovať za rovnaké, preto si ďalší záznam do tabuľky nepridá. Z toho taktiež vyplýva, že v prípade, kedy klient nie je prihlásený ku skupine 228.0.0.1 ale ku 227.0.0.1 by switch posielal dáta aj pre túto 228.0.0.1 multicastovú adresu. Rozposlal by tieto dáta na všetky rozhrania, ktoré by korešpondovali s MAC adresou 0100.5e00.0001, čo je však nežiadúce. Preto je vhodné voliť v takýchto situáciách IP multicastové adresy odlišujúce sa až od druhého oktetu (s výnimkou prvého bitu druhého oktetu). Problematiku som bližšie rozviedol v teoretickej časti diplomovej práce.

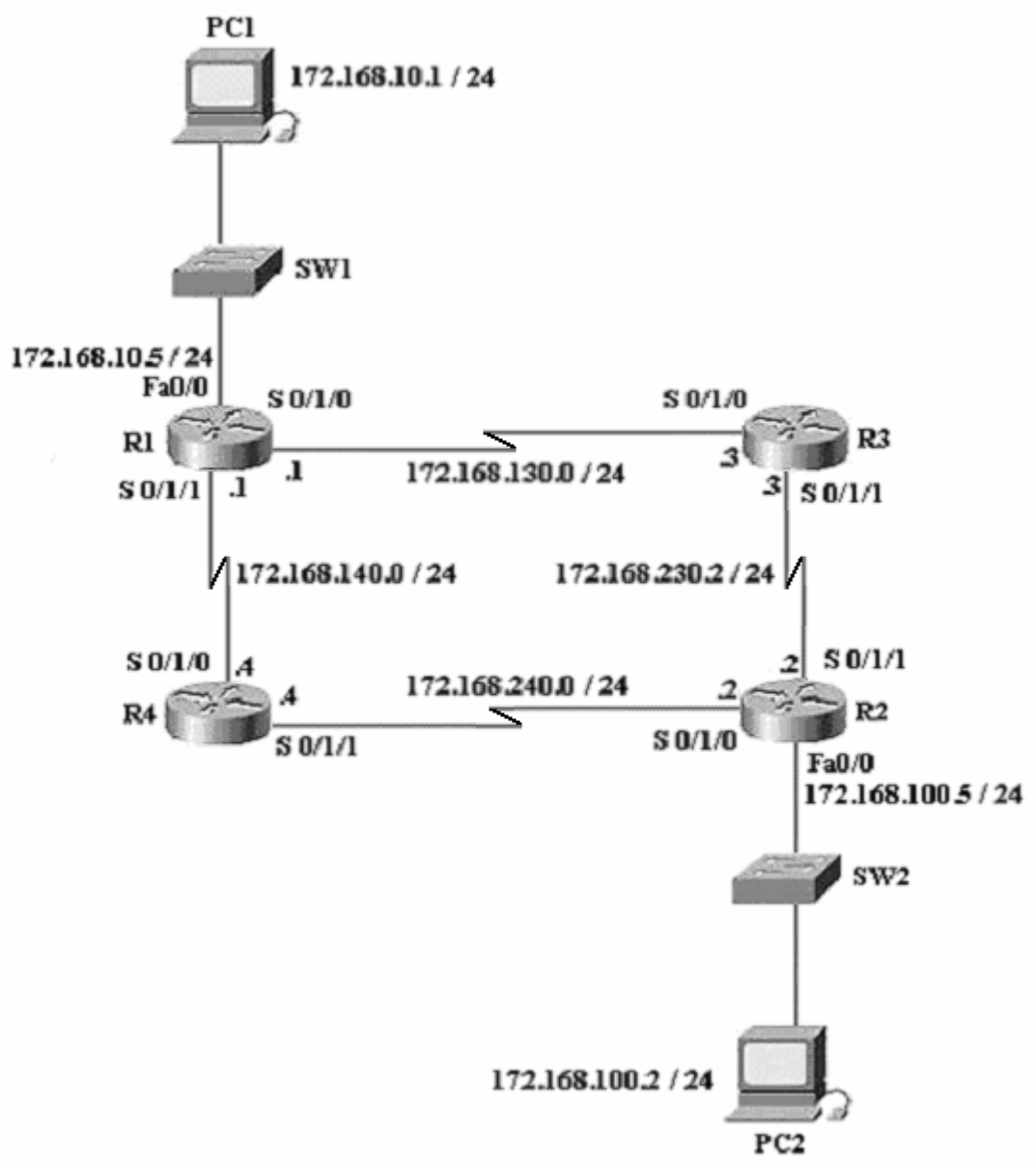
Zhrnutie:

Zapojenie v tomto teste možno považovať za neštandardné, ale práve tento fakt som týmto testom chcel dosiahnuť. Snažil som sa preskúmať situácie, ktoré nie sú typické a ani výsledok fungovania nie je priamočiary. Testovaný je predovšetkým IGMP snooping. IGMP snooping je

obvykle využívaný v smere od routra cez switch k jednotlivým príjemcom. V uvedenom teste som testoval, či IGMP snooping funguje aj v opačnom smere, teda keď požadujeme, aby nahliadal do dát v smere od počítačov k routru. V tomto prípade, kedy sa prihlasuje klient k viacerým z routrov však nemá dostatok informácií aby sa rozhodol. Preto klienta prihlási k obidvom routrom. Ďalšou problematikou uvedenou v teste bolo mapovanie multicastových skupín. Pri tomto mapovaní dochádza k určitej strate informácií, a v teste je uvedený prípad, kedy to môže spôsobovať problémy.

6.4 Test č. 4

Zapojenie v teste č. 4 pozostáva zo 4 routrov, 2 switchov a 2 počítačov.
Zapojenie č.4 je na obrázku 21.



Obrázok 21: Štvrté testované zapojenie.

Cieľ testovania:

Prihlasovanie klienta do skupiny, vysielanie IGMP paketov na rôzne adresy, pasívne a aktívne zisťovanie multicastu v LAN sieti.

Použité programy:

- SendIGMP
- CaptureIGMP

Priebeh testovania:

Vo štvrtom zapojení sa oproti predchádzajúcim zapojeniam snažím o testovanie bežnej LAN siete, ktorá obsahuje viac počítačov a taktiež viac routrov. Tak ako som už uviedol, IGMP paket, ktorý vysielam s úmyslom prihlásiť sa k určitej skupine, musím poslať na danú multicastovú adresu. Ako cieľová adresa bude uvedená daná multicastová adresa. V prvom zapojení som uviedol, že vyslanie tohoto paketu na inú multicastovú skupinu routra problém nespôsobí. Paket router príjme ako každý iný, prečíta z tela paketu, kde sa nachádza skutočná požadovaná multicastová skupina. Situáciu by mohlo zmeniť vyslanie takéhoto paketu na adresu 224.0.0.1, ktorá reprezentuje adresu všetkých multicastových zariadení na sieti. V tejto práci som už uviedol situáciu, kedy by toto vyslanie na adresu 224.0.0.1 mohlo získať na význame. Ako príklad by som uviedol firmu s desiatkami až stovkami počítačov a routrov. V sieti by bola podpora multicastu a bolo by požadované, aby každý z počítačov prijal rovnaké dáta. Bolo by samozrejme možné jednotlivo z každého počítača vyslať IGMP paket so zámerom prihlásiť sa ku skupine. Taktiež by potom tento príklad mohol poslúžiť ako hromadné odhlásenie zo skupiny, kedy by boli všetky (alebo väčšia časť) počítačov prihlásených a bolo by vhodné ich všetky čo najefektívnejšie odhlásiť. Vyslanie IGMP paketu s prihlásením, alebo odhlásením na adresu 224.0.0.1 je predmetom testovania tejto kapitoly.

Ďalšou časťou je aktívny a pasívny listening, aj za účelom ktorým bol vytvorený program CaptureIGMP.

Po nakonfigurovaní zapojenia, je nutné povoliť multicast a taktiež na jednotlivých rozhraniach povoliť PIM protokol. Konkrétna podoba konfigurácie je uvedená v prílohe 4. V tomto prípade sa jedná o PIM - DM. Routre, ktoré tento multicast, alebo protokol nemajú nastavené, nebudú multicastový traffic prenášať. Samozrejme ani neumožňujú prihlasovanie klientov do skupín. Prezentované zapojenie je vhodné pre testovanie podobných situácií. Router R3 nemá zapnutú multicastovú podporu. Naproti tomu, ostatné routre, R1, R2 a R3 sú multicastové. V prípade, že sú dáta vysielané z routra R1 do R2, existuje multicastová cesta medzi týmito dvoma routrami. Router R3 bude z cesty vynechaný. Príslušný protokol vytvorí multicastový strom, na základe ktorého sú od zdroja multicastových dát posielané tieto dáta do listov daného stromu.

Nasleduje testovanie posielania IGMP paketu na adresu 224.0.0.1. Tento paket je typu *Membership Report*, jedná sa teda o prihlásenie klienta k skupine. Paket je z počítača PC1 vyslaný programom SendIGMP na adresu 224.0.0.1 v tele s adresou 225.0.0.25. Na všetkých routroch sú

povolené ladiace režimy, čím je možné sledovať ako routre zaregistrujú tento paket. Na PC2 je pustený program CaptureIGMP za účelom zachytenia akýchkoľvek prichádzajúcich IGMP paketov. Na prvom routry R1 podľa očakávania dostáva router paket, ktorým prihlasuje klienta do skupiny. Všetky ostatné routre však tento paket už nedostávajú, ako ukazujú ich ladiace výpisy. Ani počítač PC2 s programom CaptureIGMP tento paket nezaznamenal. Ďalším krokom je overenie situácie tým, že na každom z routrov overíme skupiny, ku ktorým je prihlásený. Každý z routrov implicitne obsahuje skupinu 224.0.1.40, avšak skupina 225.0.0.25 je uvedená len na routry R1. Podobne, vyslaním *Leave group* správy IGMP paketom na adresu 224.0.0.1 klienta odhlási len z príslušného routra. Ostatné routre túto správu ani nezaregistrujú, takže hromadné odhlásenie je týmto spôsobom nemožné. V prvom teste som uviedol, že IGMP paket vyslaný na inú multicastovú skupinu, než je uvedená v tele paketu, problém nespôsobuje. V 4. teste je vidieť poslanie paketu na adresu 224.0.0.1, čo taktiež problém nevyvolal. Z uvedeného je možné usudzovať, že IGMP paket vyslaný z klienta je zachytený prvým routrom, bez ohľadu na cieľovú adresu. Router následne tento paket spracuje. Iná situácia nastáva, keď paket vyšleme miesto multicastovej adresy na unicastovú. Cieľovou adresou je napríklad adresa niektorého z rozhraní vzdialenejšieho routra. V tomto prípade sa nejedná o router R1, ale niektorý iný router. Vzhľadom k tomu vyšlem paket s cieľovou adresou 172.168.100.5 na router R2. Router R2 na danú situáciu zareaguje nasledovne.

```
*Apr 19 14:12:29.631: IP: s=172.168.10.1 (Serial0/1/0), d=172.168.100.5,
len 28, rcvd 4
*Apr 19 14:12:29.635: IGMP(0): Received v2 Report on Serial0/1/0 from
172.168.10.1 for 227.1.1.1
*Apr 19 14:12:29.635: IGMP(0): Received Group record for group 227.1.1.1,
mode 2 from 172.168.10.1 for 0 sources
```

Tento paket je na routry R1 považovaný za unicastový paket, preto ho na základe svojich smerovacích tabuliek pošle „správnym smerom“ až dorazí k routru R2. Router R2 rozpozná IGMP paket a uskutoční prihlásenie klienta až na tomto routry.

V úvode tejto kapitoly som uvádzal možnú situáciu, kedy by bolo žiadúce z danej LAN siete prihlásiť všetky počítače efektívnejším spôsobom. Pomocou vyslania IGMP paketu na adresu 224.0.0.1 to nefunguje, avšak je aspoň možnosť obsluhovať danú požiadavku z jedného počítača. Prihlásiť sa z tohoto počítača na všetky ostatné routre práve pomocou vyslania IGMP paketu s unicastovou IP cieľovou adresou daných routrov.

V nasledujúcich riadkoch nasleduje rozšírenie problematiky aktívneho a pasívneho listeningu. Pasívny listening možno vysvetliť ako načúvanie na sieti s úmyslom zachytiť to, čo po sieti putuje. Konkrétne pakety, ktoré sú medzi jednotlivými zariadeniami na sieti vymieňané. Pasívny listening preto nevyžaduje žiadnu aktívnu činnosť od programu, ktorý načúva. Naopak aktívny listening predpokladá aktívnu činnosť za účelom získania odpovede. Podobne pristupujem k pasívnemu, resp. aktívnemu listeningu v mojej diplomovej práci. Pasívny listening som už popísal v zapojení 2, kde

stačí program CaptureIGMP spustiť na počítači, ktorý je pripojený k multicastovému routru. Tento program načúva na pripojenom rozhraní a informuje užívateľa o tom, či je multicast na danom routry spustený. Spustením programu na routry, ktorý nie je multicastový, nie je možné zistiť, či daná sieť podporuje multicast. Toto tvrdenie som overil nasledujúcim spôsobom. Routre R1, R2 a R3 sú multicastové. Router R2 mal multicast vypnutý. Po spustení programu CaptureIGMP na počítači PC2 som sa snažil získať nejaké informácie o tom, či je na sieti multicast. Po pripojení klienta na počítači PC1 k multicastovej skupine 228.0.8.8 sa tento klient prihlásil do skupiny. Program na PC2 však na túto situáciu žiadnym spôsobom nezareagoval.

Za zmienku stojí v tejto situácii zobrazenie multicastových smerovacích tabuliek niektorých routrov.

R3# **show ip mroute**

```
IP Multicast Routing Table
(*, 228.0.8.8), 00:01:19/stopped, RP 0.0.0.0, flags: D
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Serial0/1/1, Forward/Dense, 00:01:19/00:00:00
    Serial0/1/0, Forward/Dense, 00:01:19/00:00:00
(172.168.100.2, 228.0.8.8), 00:01:19/00:01:44, flags: PT
  Incoming interface: Serial0/1/1, RPF nbr 172.168.230.2
  Outgoing interface list:
    Serial0/1/0, Prune/Dense, 00:01:18/00:01:44, A
(*, 224.0.1.40), 01:12:07/stopped, RP 0.0.0.0, flags: DCL
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Serial0/1/1, Forward/Dense, 01:11:39/00:00:00
    Serial0/1/0, Forward/Dense, 01:12:07/00:00:00
```

Uvedený výpis ukazuje už známu skutočnosť, že multicastový router je prihlásený k skupine 224.0.1.40. Okrem toho sa tu však nachádzajú aj v tejto chvíli zaujímavejšie informácie. Ako možno vidieť, zdroj multicastových dát pre multicastovú skupinu 228.0.8.8 je umiestnený na adrese 172.168.100.2 a RPF značí *Reverse Path Forwarding*. Táto hodnota poskytuje informáciu o tom, z akej adresy, resp. z ktorého rozhrania dostáva router informácie. RPF som už popísal v teoretickej časti, kde som uviedol, že hodnota je získavaná z unicastového smerovacieho protokolu. V prípade, že router dostáva dáta z iného než RPF rozhrania, tieto dáta zahodí. V tomto prípade je RPF pre router R3 so zdrojom multicastových dát z PC2 na rozhraní routru R2 s adresou 172.168.230.2.

Podobná situácia je na routry R4.

R4# **show ip mroute**

```
IP Multicast Routing Table
(*, 228.0.8.8), 00:00:39/stopped, RP 0.0.0.0, flags: D
  Incoming interface: Null, RPF nbr 0.0.0.0
```

```
  Outgoing interface list:
    Serial0/1/1, Forward/Dense, 00:00:39/00:00:00
    Serial0/1/0, Forward/Dense, 00:00:39/00:00:00
(172.168.100.2, 228.0.8.8), 00:00:39/00:02:20, flags: PT
  Incoming interface: Serial0/1/1, RPF nbr 172.168.240.2
  Outgoing interface list:
    Serial0/1/0, Prune/Dense, 00:00:39/00:02:29
(*, 224.0.1.40), 01:21:32/stopped, RP 0.0.0.0, flags: DCL
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Serial0/1/1, Forward/Dense, 01:20:04/00:00:00
    Serial0/1/0, Forward/Dense, 01:21:32/00:00:00
```

RPF informuje o tom, že dáta sú od zdroja multicastových dát posielané cez router R2 na rozhraní 172.168.240.2.

Router R2 je priamo pripojený k počítaču PC2, z ktorého prúdia multicastové dáta, preto má v RPF nulovú hodnotu.

R2# **show ip mroute**

```
IP Multicast Routing Table
(*, 228.0.8.8), 00:00:45/stopped, RP 0.0.0.0, flags: D
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Serial0/1/1, Forward/Dense, 00:00:45/00:00:00
    Serial0/1/0, Forward/Dense, 00:00:45/00:00:00
(172.168.100.2, 228.0.8.8), 00:00:45/00:02:20, flags: PT
  Incoming interface: FastEthernet0/1, RPF nbr 0.0.0.0
  Outgoing interface list:
    Serial0/1/0, Prune/Dense, 00:00:45/00:02:17
    Serial0/1/1, Prune/Dense, 00:00:44/00:02:15
(*, 225.0.0.22), 00:02:22/00:00:37, RP 0.0.0.0, flags: D
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Serial0/1/1, Forward/Dense, 00:02:22/00:00:00
    Serial0/1/0, Forward/Dense, 00:02:22/00:00:00
```

Z týchto informácii o RPF je následne vytváraný multicastový strom, na základe ktorého putujú multicastové dáta od zdroja k jednotlivým príjemcom.

Aktívny listening značí aktivitu zo strany užívateľa. Vyslaním IGMP paketu so správou *Membership Query* na adresu, ku ktorej sú prihlásené viaceré routre, prinúti router odpovedať. Táto odpoveď je adresovaná práve na danú multicastovú skupinu.

Zhrnutie:

Dané testovacie zapojenie je ideálne pre testovanie funkcií multicastu, ktoré je možné nájsť v obecne ľubovolnej LAN sieti. K dispozícii sú počítače pripojené k routrom, z ktorých niektoré môžu podporovať multicast, iné nie. Program, ktorý je v tejto časti primárne prezentovaný je program CaptureIGMP. Ako som už uviedol, pripojením k routru bez multicastu program zlyháva, nakoľko nedokáže zistiť multicast, ktorý sa „odohráva“ na inom multicastovom routry v sieti. Otestoval som tiež vysielanie IGMP paketov na inú multicastovú skupinu, než je multicastová skupina, na ktorú som sa chcel pripojiť. Tento fakt však žiaden problém nespôsobuje. IGMP paket je typicky posielaný na multicastovú skupinu. Za veľmi zaujímavú možnosť považujem vyslanie IGMP paketu na unicastovú adresu. Routre, cez ktoré IGMP paket prechádza ho za IGMP paket nepovažujú a prepošlú ho na základe vlastnej smerovacej tabuľky. Keď IGMP paket s unicastovou cieľovou adresou dorazí na miesto, až tu je rozpoznaný ako IGMP paket a spracovaný.

7 Záver

Teoretickú časť diplomovej práce som zameral hlavne na vysvetlenie multicastu. Taktiež som sa snažil o načrtnutie rozdielu multicastu od klasickej unicastovej komunikácii a broadcastom. Predovšetkým jeho výhody, ale aj problémy, kvôli ktorým nie je tak široko rozšírený, ako by si možno zaslúžil. Taktiež som v teoretickej časti rozobral protokol IGMP, predovšetkým ktorým sa táto práca zaoberá. Protokol IGMP úzko súvisí s protokolom IP, preto som popísal aj tento protokol. Neodmysliteľnou súčasťou multicastu sú multicastové smerovacie protokoly, na ktoré som tiež poukázal.

V praktickej časti som sa snažil o testovanie multicastu v dvoch rovinách. Na jednej strane to bolo vysvetlenie práce multicastu, výhody a problémy, ktoré sa vyskytujú. Na druhej strane, na Internete existuje veľké množstvo informácii týkajúcich sa multicastu. Chcel som poukázať na rôzne neštandardné situácie, ktoré nie sú bežne pri článkoch o multicaste prezentované. Ďalšou výstupnou hodnotou práce sú programy, ktoré som za účelom testovania vytvoril. Dva programy SendMcast a GetMcast sú ideálne k prezentácii multicastu a jeho výhod. Využívajú BSD sockety, ktoré multicast značne podporujú. Ďalšie programy sú SendIGMP a GetIGMP. Program SendIGMP je vhodný na vytváranie IGMP paketov, ktoré boli pri testovaní nevyhnutné. Je možné vytvárať rôzne správy IGMP protokolu, čo je pri testovaní veľmi užitočné. Programom CaptureIGMP je možné prijímať IGMP pakety, ktoré putujú po sieti. Tomuto programu by som pripísal najväčší význam z praktického hľadiska. Nakoľko umožňuje, za určitých predpokladov spomenutých v praktickej časti zistiť, či je v sieti poskytovaný multicast.

Medzi štandardné testovacie úlohy, ktoré som realizoval, radím vysielanie dát z jedného zdroja, jeden krát, viacerým príjemcom. Ďalšou typickou vlastnosťou multicastu, je záležitosť týkajúca sa switcha - IGMP snooping. Tento IGMP snooping umožňuje switchu podobné rozhodovanie a tým smerovanie IGMP paketov ako dokáže router. Tu som však uviedol aj menej typické situácie, kedy sa klient prihlasoval k skupine a od IGMP snoopingu som očakával spomenuté rozhodovanie, ktoré sa avšak nekonalo. Switch nedisponoval potrebnými informáciami. Medzi ďalšie neštandardné situácie uvediem vyslanie IGMP paketu na rôzne multicastové skupiny. Výsledok testu naznačil, že toto vysielanie nevplýva na funkčnosť multicastu. Za najväčší prínos tejto práce považujem posledný realizovaný test spolu s využitým programom CaptureIGMP, pomocou ktorého je možné detekovať multicast na sieti. Taktiež je možné realizovať príjem jednotlivých IGMP paketov. Tieto pakety sú programom CaptureIGMP priblížené pohľadom na ich štruktúru. Týmto upieram pozornosť najmä na typ IGMP správy a položku obsahujúcu multicastové skupiny, ktorými je možné bližšie určiť, čo sa z pohľadu multicastu na danej sieti deje.

Túto diplomovú prácu možno považovať za úvod do problematiky multicastu. Multicast je však rozsiahli a ponúka značné množstvo možností pre testovanie. K nim radím prácu s protokolom

PIM. Tento multicastový smerovací protokol je popísaný v teoretickej časti diplomovej práce. Obsahuje správy *Join/Prune*, pomocou ktorých sa prihlasujú, resp. odhlasujú samotné routre. Na základe týchto správ je potom zostavený príslušný strom. Táto problematika ponúka značné možnosti testovania. Za predpokladu, že je vytvorená správa protokolu PIM, je možné ju vysielať z počítača a týmto testovať možnosti práce s týmto protokolom.

Cisco routre, na ktorých som multicast testoval podporujú len protokoly PIM a DVMRP. Ďalšie možnosti testovania sa ponúkajú pri práci s ostatnými multicastovými protokolmi, prevažne protokolom MOSPF a protokolom CBT. Ďalšou samostatnou oblasťou testovania je aplikačný multicast. Tento druh multicastu je taktiež popísaný v práci. Funguje na princípe multicastu, ale nepredpokladá podporu zo strany routrov, naopak, túto činnosť preberajú klienti na počítačoch. Títo klienti samotní sa o všetky záležitostí starajú.

Záverom by som dodal, že moja práca mala za úlohu ukázať, aké výhody môže multicast priniesť užívateľom a organizáciám v oblasti počítačových sietí, keďže multicast stále nie je bežne rozšírený. Na druhej strane práca poukázala aj na rôzne situácie spojené s problematikou multicastu.

Literatúra

[1] Cisco Systems, Inc.: *Internet Protocol Multicast* [online]. Posledná modifikácia: 2006-10-12. [cit. 2007-12-17]. Dostupné na URL: <http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/ipmulti.htm>.

[2] Daud, I.: *Data communication* [online]. [cit. 2007-12-17]. Dostupné na URL: <http://www.niit.edu.pk/~imrandaud/CC201/All%20Lecture.ppt>.

[3] Odvárka, P.: *Formát hlavičky IPv4 protokolu, ARP protokol* [online]. Posledná modifikácia: <2000-12-22> Dostupné na URL: <http://www.svetsiti.cz/view.asp?rubrika=Tutorialy&temaID=1&clanekID=13>.

[4] Pištěk, P.: *Multicast: skupinové vysielanie* [online]. Posledná modifikácia: 2007-12-13. [cit. 2007-12-17]. Dostupné na URL: <http://www.ics.muni.cz/zpravodaj/articles/134.html>.

[5] Cisco Systems, Inc.: *Guidelines for Enterprise IP Multicast Address Allocation* [online]. [cit. 2007-12-17]. Dostupné na URL: <http://www.cisco.com/en/US/tech/tk828/technologies_white_paper09186a00802d4643.shtml>.

[6] Miller, C. K.: *Multicast Networking and Applications*. Addison-Wesley 1998. ISBN 0201309793.

[7] H3C Technologies Co., Limited: *Introduction to Multicast* [online]. [cit. 2007-12-17]. Dostupné na URL: <http://www.h3c.com/portal/res/200706/18/20070618_113552_Mulitcast%20Overview_201278_57_0.pdf>.

[8] Deering, S.E.: *Host Extensions for IP Multicasting*, IETF RFC 1112 IGMP v1 [online]. [cit. 2007-12-20]. Dostupné na URL: <ftp://ftp.isi.edu/in-notes/rfc1112.txt>.

[9] Fenner, W.: *Internet Group Management Protocol, Version 2* [online]. [cit. 2007-12-20]. Dostupné na URL: <ftp://ftp.isi.edu/in-notes/rfc2236.txt>.

[10] Cain, B., Deering, S., Kouvelas, I., Fenner, B., Thyagarajan, A.*: Internet Group Management Protocol, Versoon 3* [online]. [cit. 2007-12-20]. Dostupné na URL: <ftp://ftp.isi.edu/in-notes/rfc3376.txt>.

[11] Filip, O.: *Úvod do IP multicastu* [online]. Posledná modifikácia: 2004-15-24. [cit. 2008-05-02]. Dostupné na URL: < http://www.lupa.cz/clanky/uvod-do-ip-multicastu-dil-paty/>.

[12] Dostálek, L.: *IP protokol* [online]. [cit. 2008-05-02]. Dostupné na URL: <http://www.cpress.cz/knihy/tcp-ip-bezp/CD-0x/5.html>.

[13] Mojžíšek, P.: *Multicast na Internetu* [online]. Posledná modifikácia: 2002-3-1. [cit. 2007-12-20]. Dostupné na URL: <http://www.isdn.cz/clanek.php?cid=3626>.

[14] Filip, O.: *Úvod do multicastu* [online]. Posledná modifikácia: 2004-10-29. [cit. 2007-12-20]. Dostupné na URL: <http://www.lupa.cz/clanky/uvod-do-ip-multicastu-dil-treti/>.

[15] Waitzman, D., Partridge, C., Deering, S.: *Distance Vector Multicast Routing Protocol* [online]. [cit. 2007-12-20]. Dostupné na URL: <http://www.ietf.org/rfc/rfc1075.txt>.

[16] Adams, A., Nicholas, J., Siadak, W.: *Protocol Independent Multicast – Dense Mode (PIM-DM)*: Protocol Specifiction [online]. [cit. 2007-12-20]. Dostupné na URL: <http://www.ietf.org/rfc/rfc3973.txt>.

[17] Cisco Systems, Inc.: *Multicasting in IP and Apple Talk Networks* [online]. Posledná modifikácia: 2006-10-12. [cit. 2007-12-28]. Dostupné na URL: <http://www.cisco.com/univercd/cc/td/doc/cisintwk/idg4/nd2024.htm>.

[18] Estrin, D., Ferinacci, D., Helmy, A., Thales, D., Deering, S., Handley, M., Jacobson, V., Liu, C., Sharma, P., Wei, L.: *Protocol Independent Multicast - Sparse Mode (PIM-SM)*: *Protocol Specifiaciton* [online]. [cit. 2007-12-20]. Dostupné na URL: <http://www.ietf.org/rfc/rfc2362.txt>.

[19] Banerjee, S., Bhattacharjee, B., Kommareddy, Ch.: *Scalable Application Layer Multicast* [online]. [cit. 2008-04-02]. Dostupné na URL: <http://dslab.csie.ncu.edu.tw/94html/paper/pdf/Scalable%20Application%20Layer%20Multicast.pdf>.

[20] Stevens, W., Fenner, B., Rudoff, A.: *Network Programming Volume 1, Third Edition: The Sockets Networking*. Addison-Wesley 2003. ISBN 0131411551.

Zoznam príloh

Príloha 1. Konfigurácia pre zapojenie č. 1

R:

```
hostname R
interface FastEthernet0/0
  ip address 172.168.10.5 255.255.255.0
  no shutdown
```

PC1: nastavenie IP adresy na 172.168.10.1 a masky 255.255.255.0

PC2: nastavenie IP adresy na 172.168.10.2 a masky 255.255.255.0

PC3: nastavenie IP adresy na 172.168.10.3 a masky 255.255.255.0

Príloha 2. Konfigurácia pre zapojenie č. 2

R:

```
hostname R
ip multicast-routing
interface FastEthernet0/0
  ip address 172.168.10.5 255.255.255.0
  no shutdown
  ip pim dense-mode
interface FastEthernet0/1
  ip address 172.168.20.5 255.255.255.0
  ip pim dense-mode
  no shutdown
router eigrp 1
  network 172.168.0.0
```

PC1: nastavenie IP adresy na 172.168.10.1 a masky 255.255.255.0

PC2: nastavenie IP adresy na 172.168.20.2 a masky 255.255.255.0

PC3: nastavenie IP adresy na 172.168.20.3 a masky 255.255.255.0

Príloha 3. Konfigurácia pre zapojenie č. 3

R1:

```
hostname R1
ip multicast-routing
interface FastEthernet0/0
```

```
  ip address 172.168.10.5 255.255.255.0
  ip pim dense-mode
  no shutdown
```

R2:

```
hostname R2
ip multicast-routing
interface FastEthernet0/0
  ip address 172.168.10.6 255.255.255.0
  ip pim dense-mode
  no shutdown
```

PC: nastavenie IP adresy na 172.168.10.1 a masky 255.255.255.0

Príloha 4. Konfigurácia pre zapojenie č. 4

R1:

```
hostname R1
ip multicast-routing
interface FastEthernet0/0
  ip address 172.168.10.5 255.255.255.0
  ip pim dense-mode
  no shutdown
interface Serial0/1/0
  bandwidth 64
  ip address 172.168.130.1 255.255.255.0
  clock rate 64000
  ip pim dense-mode
  no shutdown
interface Serial0/1/1
  bandwidth 64
  ip address 172.168.140.1 255.255.255.0
  ip pim dense-mode
  no shutdown
router eigrp 1
  network 172.168.0.0
```

R2:

```
hostname R2
```

```
ip multicast-routing
interface FastEthernet0/0
  ip address 172.168.100.5 255.255.255.0
  ip pim dense-mode
  no shutdown
interface Serial0/1/0
  bandwidth 64
  ip address 172.168.240.2 255.255.255.0
  clock rate 6400
  no shutdown
interface Serial0/1/1
  bandwidth 128
  ip address 172.168.230.2 255.255.255.0
  clock rate 128000
  ip pim dense-mode
  no shutdown
router eigrp 1
  network 172.168.0.0
```

R3:

```
hostname R3
ip multicast-routing
interface Serial0/1/0
  bandwidth 64
  ip address 172.168.130.3 255.255.255.0
  clock rate 64000
  ip pim dense-mode
  no shutdown
interface Serial0/1/1
  bandwidth 128
  ip address 172.168.230.3 255.255.255.0
  ip pim dense-mode
  no shutdown
router eigrp 1
  network 172.168.0.0
```

R4:

```
hostname R4
ip multicast-routing
interface Serial0/1/0
  bandwidth 64
  ip address 172.168.140.4 255.255.255.0
  clock rate 64000
  ip pim dense-mode
  no shutdown
interface Serial0/1/1
  bandwidth 128
  ip address 172.168.240.4 255.255.255.0
  ip pim dense-mode
  no shutdown
router eigrp 1
  network 172.168.0.0
```

PC1: nastavenie IP adresy na 172.168.10.1 a masky 255.255.255.0

PC2: nastavenie IP adresy na 172.168.100.2 a masky 255.255.255.0

Príloha 5. CD