UNIVERZITA KOMENSKÉHO V BRATISLAVE

FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

KATEDRA INFORMATIKY

QOS A MULTICASTY V PROSTREDÍ OSPF

MIROSLAV MAJDAN

2007

QOS A MULTICASTY V PROSTREDÍ OSPF

DIPLOMOVÁ PRÁCA

MIROSLAV MAJDAN

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY
KATEDRA INFORMATIKY

Študijný odbor: informatika, 2508800

Vedúci diplomovej práce: Ing. Ladislav Ivančík

BRATISLAVA 2007

Zadanie

Preskúmať dnešné možnosti technológií QoS a multicast. Zamerať sa na prostredia s routovacím protokolom OSPF. Overiť praktické využitie týchto technológií buď pomocou vlastnej implementácie, alebo použitím existujúcich nástrojov.

Čestné vyhlásenie

Vyhlasujem, že som diplomovú prácu riešil samostatne, s použitím uvedených prameňov a literatúry pod vedením môjho školiteľa.

Bratislava, 2 máj 2007 .....

Miroslav Majdan

Poďakovanie

Ďakujem svojmu školiteľovi Ing. Ladislavovi Ivančíkovi za odborné rady, vedenie a cenné pripomienky k mojej práci a svojim blízkym za podporu.

Abstrakt

Majdan, Miroslav: QoS a Multicasty v prostredí OSPF. Diplomová práca, Univerzita Komenského. Fakulta matematiky, fyziky a informatiky, Katedra informatiky. Vedúci diplomovej práce: Ing. Ladislav Ivančík. Bratislava : Fakulta matematiky, fyziky a informatiky UK, 2007.

Práca sa zaoberá technológiami QoS a multicast so zameraním na ich použitie v sieťach s routovacím protokolom OSPF. Popisuje teoretické základy samotného routovania, QoS routovania a multicastového routovania. Poskytuje prehľad dnešných techník, používaných na zabezpečenie QoS (DiffServ, IntServ), prehľad možností multicastového prenosu dát a popis protokolu OSPF vrátane jeho QoS a multicastových rozšírení. Podáva reálny pohľad na stav týchto technológií a ich použiteľnosť v existujúcich sieťach. Nakoniec ukazuje možné praktické využitie týchto technológií v jednoduchej sieti, kde využíva vlastnú implementáciu protokolu MOSPF a službu DiffServ implementovanú v existujúcich linuxových nástrojoch. V záverečných testoch ukazuje význam a výhody týchto technológií.

Kľúčové slová: routovanie, QoS, multicast, OSPF.

Obsah

Slovník použitých skratiek

AS – Autonomous System

BDR – Backup Designated Router

bps – bity za sekundu

CGMP – Cisco Group Management Protocol

DiffServ – Differentiated Services

DR – Designated Router

DSCP – DiffServ Codepoint

DVMRP – Distance Vector Multicast Routing Protocol

EGP – Exterior Getaway Protocol

IGMP – Internet Group Management Protocol

IGP – Interior Getaway Protocol

IntServ – Integrated Services

IP – Internet Protocol

kbps – kilobity za sekundu

LAN – Local Area Network

MOSPF – Multicast OSPF

OSPF – Open Shortest Path First

QoS – Quality Of Service

QOSPF – QoS OSPF

RSVP – ReSerVation Protocol

SLA – Service Level Agreement

SPF – Shortest Path First

TCA – Traffic Conditioning Agreement

TOS – Type Of Service

TTL – Time To Live

1 Úvod

Dnešné siete pracujú na rýchlostiach rádovo megabity/s. Užívatelia chcú používať moderné multimediálne aplikácie, ktoré sú ale stále veľmi náročné na objem prenesených dát. Typickým príkladom sú videokonferencie. Umožňujú plnohodnotnú komunikáciu medzi ľuďmi, ktorý sa nachádzajú na rôznych miestach. Spoločnosti po nich čím ďalej tým viac siahajú v prípade, ak je fyzický presun nemožný alebo finančne a časovo nevýhodnejší. V dnešnej dobe najpoužívanejší štandard na prenos videa je H.323. Pri tomto štandarde je odporúčaná prenosová kapacita na jeden tok videa 384kbs.

Sieťové spojenie pri videokonferenciách má väčšinou topológiu stromu, kde koreň stromu je server, ktorý vysiela dáta a listy sú účastníci videokonferencie, ktorý ich prijímajú. Takáto situácia je napríklad pri internetovom vysielaní nejakej televízie, vysielanie prednášok na fakulte atď. Pokiaľ chcú aktívne komunikovať všetci účastníci medzi sebou, jedným z možných riešení je vytvorenie *n* takýchto stromov, z ktorých každý koreň bude predstavovať jedného účastníka videokonferencie. Iným riešením je prenášať dáta po vetvách stromu obojstranne.

1.1 Situácia

Uvažujme videokonferenciu s jedným serverom a 10 pasívnymi účastníkmi. Pri použití kódovania H.323 je zaťaženie sieťového pripojenia serveru niekoľko Mbs. Toto nie je veľký problém v prípade, že sa videokonferencia koná v rámci jednej LAN siete. Problém vzniká, ak je server umiestnený v inej firme ako ostatní účastníci. Pripojenie do internetu s kapacitou niekoľko Mbs môže byť pre niektoré firmy finančne neúnosné.

Iný problém je prioritizácia. Pri pripojení ďalších účastníkov môže ľahko dôjsť k zahlteniu pripojenia, čo môže vyvolať veľké problémy v prípade potreby prenosu kritických dát. Takisto ak sa po pripojení ďalších účastníkov prekročí prenosová kapacita linky, bude to mať negatívny dopad na kvalitu spojenia existujúcich

účastníkov, čo môže v konečnom dôsledku viesť k tomu, že celá videokonferencia bude nezrozumiteľná - a teda zbytočná - ale vyťaží celú prenosovú kapacitu.

1.1.1 Riešenie – multicast

V našej situácii potrebujeme preniesť rovnaké dáta od jednej k viacerým staniciam. V prípade klasického (unicastového) riešenia musí vysielacia stanica vytvoriť rovnakú kópiu pre každú cieľovú stanicu a tieto dáta poslať každej samostatne. V prípade multicastového prenosu zdrojová stanica pošle jednu kópiu dát a ako príjemcu uvedie *skupinu*, v ktorej sa nachádzajú cieľové stanice. Medzi zdrojovou a cieľovými stanicami sa vytvorí strom. Po každej hrane stromu sa prenáša len jedna kópia dát. Dáta sa duplikujú len v miestach, kde sa strom rozvetvuje.

1.1.2 Riešenie – QoS

Väčšina dnes zapojených aktívnych sieťových prvkov pristupuje ku všetkým dátam rovnako. V našej situácii (videokonferencia) by sa nám však hodilo, keby to tak nebolo. Chceme aby sa inak (prioritne) pristupovalo napr. k dátam patriacim videokonferencii a dátam webových služieb. Takúto možnosť poskytuje QoS.

QoS je množina požiadaviek na dáta, ktoré majú byť splnené pri ich prenose sieťou [RFC2386].

V súvislosti s QoS sa často spomína aj rezervácia prostriedkov v sieti. Tieto prostriedky sa definujú pomocou QoS parametrov.

2 Teoretická časť

V tejto časti uvedieme základne pojmy a vlastnosti, ktoré sa týkajú routovania, multicastov a QoS. V ďalšom texte budeme uvažovať len o sieťach fungujúcich na základe protokolov TCP/IP.

2.1. Routovanie

Routovanie je proces hľadania ciest v počítačovej sieti, po ktorej budú prenášané dáta. Routovanie vykonávajú aktívne sieťové prvky – routre – a deje sa na sieťovej vrstve na základe IP adries. Väčšinou sa deje iba na základe cieľovej adresy. Proces samotného posielanie dát medzi sieťovými prvkami sa nazýva *smerovanie*.

Rozlišujeme dva základné druhy routovania:

1. statické – cesty pre pakety určuje administrátor, sú ním pevne určené, nemenia sa žiadnym iným spôsobom
2. dynamické – existuje routovací protokol, ktorý zabezpečuje výmenu informácií medzi routrami a tie si na základe tejto informácie vedia zistiť cesty pre jednotlivé pakety; v prípade zmeny topológie siete sa automaticky bez vonkajšieho zásahu zmenia existujúce cesty, teda sieť pokračuje ďalej s minimálnym prerušením

Existuje niekoľko routovacích protokolov. Tieto môžeme rozdeliť do dvoch skupín:

1. Interior getaway protocols (IGP) – pracujú v rámci jedného AS
2. Exterior getaway protocols (EGP) – pracujú medzi niekoľkými AS

Autonómny systém (AS) je oblasť siete, ktorej routovanie spravuje jedna entita (jednotlivec prípadne skupina ľudí, ktorý spoločne rozhodujú).

Routovacie protokoly môžu byť ďalej typu:

1. distance-vector – routre si vymieňajú medzi sebou vlastné routovacie tabuľky a na ich základe dopočítavajú vlastné
2. link-state – routre si vymieňajú informácie o celej topológii siete; každý router teda pozná celú sieť

V tejto diplomovej práci sa budeme zaoberať routovacím protokolom OSPF. Je to IGP link-state protokol a je to jeden z najrozšírenejších otvorených routovacích protokolov.

Keď sa vrátime k nášmu problému videokonferencií, routovanie má zabezpečiť nájdenie multicastového stromu pre konkrétny prenos dát a to taký, ktorý bude spĺňať QoS požiadavky na tento prenos.

2.2. QoS routovanie

Pri klasickom (*best-effort*) routovaní pristupujú routre ku každému paketu rovnako – snažia sa ho doručiť čo najlepšie vzhľadom na nejaké pevne dané metriky (väčšinou počet skokov – *hop count*).

QoS routovanie (QoS-based routing) je routovanie, kde sú cesty pre dáta zisťované na základe znalosti dostupnosti zdrojov siete a QoS požiadaviek daného prenosu. QoS požiadavky sú určené QoS parametrami (napr. prenosová šírka, latencia).

QoS routovanie sa môže diať na základe:

1. cieľovej adresy
2. cieľovej a zdrojovej adresy
3. príslušnosti k toku dát (flow-based) – pakety sa musia dať jednoznačne priradiť k nejakému toku

Routovanie na základe cieľovej adresy majú najmenšiu pamäťovú a výpočtovú náročnosť, routovanie na základe tokov najväčšiu.

QoS routovanie založené na tokoch pozostáva z 3 úloh:

1. získať potrebné QoS informácie o sieťových spojeniach v sieti a nájsť cestu pre novú požiadavku (dáta)
2. vytvoriť cestu/spojenie pre novú požiadavku
3. udržiavať (spravovať) túto cestu

Diskutabilná je otázka, ako zareagovať pri zmene charakteristík (zaťaženia) siete (napr. výskyt lepšej cesty pre už prebiehajúci tok). Ukazuje sa, že lepšie je ponechať prebiehajúce toky na svojich pôvodných cestách, zmeniť len v prípade extrémnej zmeny v sieti. Je to kvôli zachovaniu určitej stability zaťaženia siete. Ak by sme na zmenu v sieti zareagovali hneď, mohlo by to vyvolať nezadržateľný ‘lavínový efekt‘.

Jednotlivé toky môžu mať okrem svojich QoS požiadaviek nastavené aj priority. Využijú sa napríklad v prípade kedy je málo voľných prostriedkov pre dva toky, ale dosť pre jeden a je treba si vybrať ktorý.

Problémom QoS routovanae je škálovateľnosť. Rieši sa *hierarchickou agregáciou*, teda abstrahovaním väčších častí siete do jedného uzla. Pri tom, ale hrozí nebezpečenstvo veľkej nepresnosti údajov, ktoré môžeme eliminovať napríklad backtrackingom (*crankback*). Nastáva ale problém s výkonom.

QoS podobne ako normálne routovanie má dve formy:

1. intradomain – v rámci jednej domény spravovanej jednou entitou
2. interdomain – medzi doménami spravovanými rôznymi entitami

Intradomain routovanie sa rieši väčšinou link-state routovacími protokolmi. Pri interdomain routovaní by vznikali problémy so škálovateľnosťou, preto treba použiť iné riešenie.

Jedno riešenie je vychádzať zo statických QoS parametrov sietí, ktoré sú odvodené od topológie a kapacity liniek.

Druhé riešenie je, že sa zástupcovia domén zmluvne dohodnú na prideľovaní QoS parametrom spojeniam.

2.1.1 QoS parametre

QoS parametre popisujú vlastnosti resp. nároky sieťového spojenia. Takisto popisujú možnosti (kapacitu) sieťového rozhrania, spojenia alebo celej siete. Najpoužívanejie typy sú:

- prenosová šírka (bandwidth) – udávaná v bitoch/s prípadne násobkoch
- latencia (delay) – udávaná v sekundách. Poznáme dva typy:
  - end-to-end – latencia medzi koncami spojenia
  - node-to-node - latencia medzi dvoma sieťovými prvkami
- zmena latencie v čase (jitter) – problém môže nastať keď dáta dorazia v nesprávnom poradí. Musíme sa rozhodnúť či počkať na preskočený paket, alebo ho odignorovať, alebo si o ňho znova požiadať. Toto väčšinou rieši vyššia vrstva.
- cena (cost) – priraďovaná administrátorom
- stratovosť (loss rate) – dá sa štatisticky vypočítať na základe už poslaných dát

Parametre sa rozdeľujú na základe toho, ako sa vypočítavajú parametre pre celú cestu z parametrov čiastkových liniek na:

1) sčítavacie

$m(v_0, v_n) = m(v_0, v_1) + m(v_1, v_2) + \ldots + m(v_{n-1}, v_n)$

2) násobiace

$m(v_0, v_n) = m(v_0, v_1) * m(v_1, v_2) * \ldots * m(v_{n-1}, v_n)$

3) konkávne

$m(v_0, v_n) = min\,(m(v_0, v_1), m(v_1, v_2), \ldots, m(v_{n-1}, v_n))$

, kde $m(u, v)$ je hodnota QoS parametru pre cestu z vrcholu u do vrcholu v a predpokladáme, že existuje cesta $P = (v_0, v_1, \ldots, v_n)$.

Sčítacie a násobiace parametre sú ekvivalentné pri použití logaritmickej funkcie.

2.1.2 Rezervácia zdrojov

S QoS routovaním súvisí aj rezervácia zdrojov v sieti. Protokoly pre rezerváciu zdrojov (napríklad RSVP) poskytujú metódy na žiadosti a rezervácie zdrojov pre vytvárané spojenia. Neposkytujú však techniky na nájdenie cesty, ktorá vyhovuje QoS požiadavkám, preto sa musia používať súčasne s routovacími protokolmi.

Musíme mať teda definované rozhranie medzi protokolmi na rezerváciu zdrojov a QoS routovacími schémami. RSVP je navrhnuté tak, aby bolo čo najviac nezávislé na použitej routovacej schéme.

2.1.3 Riadenie prenosu (traffic control)

Riadenie prenosu je nástroj na zabezpečenie, resp. vykonávanie QoS služieb. Je to ovplyvňovanie poradia a existencie paketov na sieťovom rozhraní, väčšinou výstupnom.

Poznáme niekoľko druhov:

- *shaping* – pakety sú zdržiavané na rozhraní, aby sa udržala nejaká daná maximálna prenosová šírka
- *scheduling* – použitie rozličných front na preusporiadanie paketov
- *classifying* – rozdeľovanie paketov do tried, ku ktorým sa môže neskôr rozdielne pristupovať
- *policing* – meranie a limitovanie prenosovej šírky. Väčšinou sa deje na okrajových routroch. Siete niekedy nemusia akceptovať dáta presahujúce určitý limit.
- *dropping* – zahadzovanie paketov. Používané pri policing.

- *marking* – označovanie paketov. Používané pri classifying.

Tieto druhy nie sú presne definované a čiastočne sa funkčne prelínajú. Väčšinou sa používa ich kombinácia.

2.2 Existujúce QoS technológie

V nasledujúcej kapitole opíšeme technológie, ktoré sa v súčasnosti používajú na zabezpečenie QoS routovania.

2.2.1 TOS políčko

TOS políčko je umiestnené v IP hlavičke [RFC IP]. Má 8 bitov. Je určené na jednoduché pridelenie QoS požiadaviek jednotlivým paketom. Význam jednotlivých bitov je popísaný v tabuľke 1:

Bit	Význam
0-2	Precedence
3	0 = Normálna latencia, 1 = Nízka latencia
4	0 = Normálna prenosová rýchlosť, 1 = Vysoká p.r.
5	0 = Normálna spoľahlivosť, 1 = Vysoká spoľahlivosť
6-7	Rezervované pre budúce využitie

Tabuľka 1

Z bitov 3-5 musia byť nastavené na hodnotu 1 maximálne dva.

Tri precedence bity majú nasledovný význam:

Hodnota precedence bitov	Význam
111	Network Control

110	Internetwork Control
101	CRITIC/ECP
100	Flash Override
011	Flash
010	Immediate
001	Priority
000	Routine

Tabuľka 2

Smerom dole v tabuľke2 klesá priorita dát.

Väčšina zariadení vie rozoznať hodnoty TOS fieldu, avšak málokedy je takéto nastavenie v praxi využívané.

2.2.2 IntServ (Integrated Services)

IntServ je architektúra, ktorá umožňuje poskytovať sieti sieťové spojenia s garantovanými QoS vlastnosťami.

IntServ dáta rozdeľuje do tokov (*flows*). Týmto sú priradené QoS požiadavky. Routre si musia udržiavať informácie o tokoch, ktoré nimi prebiehajú. V sieti musia všetky routre podporovať IntServ.

Každý tok má svoju charakteristiku (*flow spec*), ktorá má dve časti:

- TSPEC (*traffic specification*) – definuje pravdepodobnú (priemernú) charakteristiku dát, ktoré budú prenášané
- RSPEC (*request specification*) – definuje požiadavky na spojenie, teda charakterizuje dáta, ktoré má byť spojenie schopné prenášať

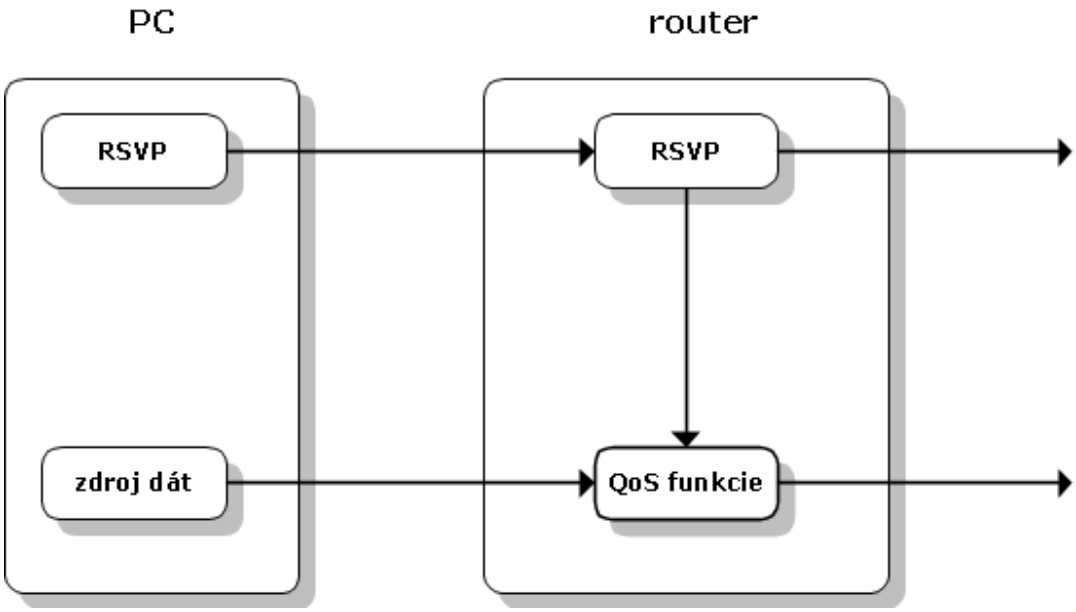
Pri požiadavke na spojenie (tok) sa môžu využiť dva druhy služieb:

1. *Guaranteed Quality of Service* (RFC 2212) – maximálne možné oneskorenie je počítané pre najhorší prípad premávky (zaťaženia) v sieti. Garantuje teda maximálnu latenciu. V požiadavke sa uvádza len TSPEC.

2. *Controlled-Load Network Element Service* (RFC 2211) - maximálne možné oneskorenie je počítané pre priemerný prípad premávky (zaťaženia) v sieti. Negarantuje maximálnu latenciu. Je určené pre aplikácie, ktoré zle znášajú zaťaženie siete. Sústreďuje sa hlavne na stratovosť a latenciu. Používa sa napr. na audio, video prenosy. V požiadavke sa uvádza TSPEC aj RSPEC.

Na zabezpečenie takýchto služieb potrebujeme mať protokol na rezerváciu zdrojov (*resource reservation protocol*). Najčastejšie používaným a prakticky jediným je protokol RSVP. IntServ ale nezávisí na použití konkrétne tohto protokolu.

Rezervácie (požiadavky na QoS služby) sú jednosmerné, na obojsmernú treba dve požiadavky. Na obrázku 1 je zachytená architektúra služby IntServ.



Obrázok 1

2.2.2.1 RSVP

RSVP je protokol používaný hlavne pri IntServ na rezervovanie zdrojov v sieti. RSVP je navrhnutý tak, aby fungoval aj v sieti, kde nepodporujú všetky zariadenia RSVP. RSVP je navrhnutý na použitie vo veľkých LAN sieťach.

Rezervácia zdrojov prebieha nasledovne:

Odosielateľ pošle *PATH* správu s *TSPEC*/RSPEC informáciami prijímateľovi. Charakteristiky môžu byť rôzne QoS parametre napríklad: veľkosť MTU, minimálna prenosová šírka, latencia. Zariadenia, ktorými požiadavka prechádza, nesmú meniť TSPEC/RSPEC dáta. PATH správa môže ale obsahovať *ADSPEC* dáta, ktoré zariadenia menia a upravujú podľa aktuálnych charakteristík siete.

Každý router si udržuje informáciu, ktorá obsahuje IP adresu predchádzajúceho routra a *session identifikátor* jednoznačne identifikujúci daný tok (zdrojová a cieľová adresa a port). Každý router modifikuje obsah PATH správy. Nastaví hodnotu IP adresy posledného routra (adresa rozhrania, ktorým poslal PATH ďalej).

Každý router, ktorý dostane PATH správu môže ju buď akceptovať alebo odmietnuť na základe TSPEC/RSPEC požiadaviek a dostupných možností.

Keď dosiahne PATH cieľ, ten pošle odpoveď *RESV* naspäť na adresu posledného routra (nie teda na adresu odosielateľa samotnej PATH správy). To zabezpečí, že paket pôjde naspäť po tej istej trase. Problémy môžu vzniknúť pri náhlej zmene topológie siete, prípadne ak sa na ceste nachádza router, ktorý nepodporuje RSVP. Keďže PATH správa je obyčajný paket s nastavenou správnou cieľovou adresou, takéto routre ho budú správne smerovať, nie je však isté, či cestou spať budú tiež tak isto smerovať správu RESV.

RESV odpoveď je paket, na základe ktorého routre, ktorými prejde, rezervujú zdroje.

Tento prístup má jeden hlavný nedostatkov. Nie je napríklad isté, či sa medzi prijatím PATH a RESV správy konkrétnym routrom nezmenila situácia v sieti. Existujú technológie, ktoré tento nedostatok riešia: napr. OPWA, ADSPEC.

To aby PATH správa bola smerovaná cestou, ktorá splňuje TSPEC/RSPEC požiadavky, zabezpečuje QoS routovanie s RSVP rozhraním.

Protokol RSVP ponúka viac: merging flows, shared reservation, multicast support atď. Vzhľadom k tomu, že sa táto architektúra už prakticky nikde nepoužíva, nebudeme sa ňou ďalej zaoberať.

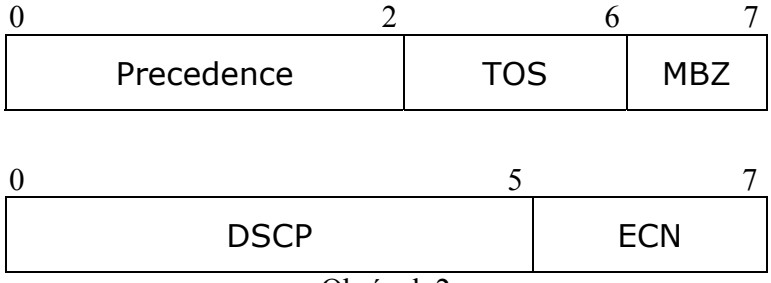
2.2.3 DiffServ (Differentiated Services)

Architektúra DiffServ sa používa na poskytovanie negarantovaných QoS parametrov. Motivácia vzniku bola snaha poskytnúť poskytovateľovi internetového pripojenia možnosť zaručiť zákazníkovi nejaké QoS služby.

Nepozerá sa na dáta ako na toky. Umožňuje rozdeliť dáta do viacerých tried (*Class of Service*). Ku každej triede môžu routre pristupovať (vzhľadom na priority) inak. DiffServ nedefinuje konkrétne pravidlá ako sa správať ku každej triede, ale poskytuje prostredie na klasifikáciu a využívanie týchto tried. Odporúča niekoľko tried, kvôli kompatibilite jednotlivých DiffServ zariadení.

2.2.3.1 DS políčko (DS field)

Architektúra DiffServ používa na rozlíšenie rozlíšenie paketov do tried (z ktorých každá vyžaduje špecifické QoS parametre) DS políčko. Nachádza sa v IP hlavičke. Nahrádza políčko TOS. Na obrázku 2 je políčko TOS v porovnaní s políčkom DS.



Obrázok 2

Z 8 bitov DS políčka sa 6 používa na určenie *DSCP* (DiffServ Codepoint) a 2 na Explicit Congestion Notification (ECN). Obsah DSCP určuje *PHB* (Per-Hop Behaviour), teda správanie sa zariadení v sieti k paketu. Z hľadiska implementácie to znamená charakteristika fronty, v ktorej budú pakety spracovávané. Napríklad priorita vrámci tejto fronty, alebo pravdepodobnosť, že pakety s konkrétnym PHB budú

vymazané ak fronta prekročí nejakú kritickú hodnotu (*drop preference – priorita vyhodenia z fronty*).

Existujú niekoľké predvolené PHB, ktoré by mali poskytovať všetky zariadenia podporujúce DiffServ:

- *default PHB* - musí obsahovať každý router. Je to tradičné best-effort routovanie. Keď router nájde neznámy DSCP, mal by sa k nemu správať ako k default PHB.
- *Class selector codepointy (xxx000)* – tieto PHB musia zachovať spätnú kompatibilitu s IP precendence bitmi TOS políčka.
- *Expedited Forwarding PHB* – malá stratovosť, latencia a zmena latencie, zaručená vysoká prenosová rýchlosť – určené napr. pre aplikácie na prenos hlasu alebo videa
- *Assured forwarding (AF) PHB* – definuje 12 rôznych BHP. Každý je definovaný číslom fronty (*x*) a prioritou vyhodenia z fronty (*y*). Oynačujú sa *AFxy*. Jednotlivé fronty sa líšia prioritou spracovávania.

IETF povoľuje akékoľvek mapovanie DSCP na PHB. Na zabezpečenie konzistencie QoS vlastností je nutné, aby bolo mapovanie konštantné. Oblasť, v rámci ktorej je mapovanie konštantné, sa nazýva *DiffServ doména*.

Pri vstupe paketov do DiffServ domény sa na okrajových routroch (*conditionery*) pakety klasifikujú a sú priradené do rozličných *behaviour agregátov*, každý identifikovaný vlastným DSCP. Toto sa deje na základe právnych zmlúv medzi jednotlivými DiffServ doménami – TCA (traffic conditioning agreement). Behaviour agregáty sú potom na jednotlivých zariadeniach mapované do konkrétnych PHB.

Z technického hľadiska je najčastejšie klasifikovanie na základe zdrojového a cieľového portu. Toto má niekoľko nevýhod:

1. fragmentácia – iba v prvom fragmente (pri IP fragmentácii) sa nachádza zdrojový a cieľový port
2. kompresia/šifrovanie – napríklad v sieťach VPN sa k informácii o portoch nedostaneme

V conditioneroch môžu byť určené pre konkrétne DSCP profily premávky (*traffic profiles*). Profily premávky určujú napr. kedy pakety budú (pri prekročení nejakého zaťaženia siete) alebo kedy sa budú označovať ktorými DSCP (tiež na základe premávky v sieti). Conditionery obsahujú väčšinou komplexné nástroje na kontrolu premávky: classifier, meter, marker, shaper/dropper.

Na conditioneroch sa deje určitá forma rezervácie zdrojov. Conditionery by mali vedieť odhadnúť aktuálne zaťaženie siete podľa objemu dát, ktoré do domény pustili, a na základe toho vedieť rozhodnúť o prijatí resp. neprijatí ďalších dát. Toto je ale veľký problém pri multicastových prenosoch, kde sa nedá dopredu odhadnúť, koľko zdrojov multicastový prenos spotrebuje. Pre multicastové dáta sa môžu použiť osobitné DSCP, ktorých prislúchajúci PHP zamedzia možnosti zahltenia siete.

Ako sa má pristupovať k dátam z inej DiffServ domény si dohadujú zúčastnené strany v tzv. SLA (service level agreement) dohode.

V komerčnom prostredí sa ujalo označenie *Bronze, Gold, Premium* premávky.

2.2.4 IntServ vs. DiffServ

Porovnanie základných vlastností IntServ a DiffServ je v nasledujúcej tabuľke:

	IntServ	**DiffServ**
škálovateľnosť	Nie	áno
garancia QoS	Áno	nie

Tabuľka 3

Pri použití IntServ aj controlled load služby poskytujú určitú dostatočnú záruku, ktorú u DiffServ nemáme. Pri zvýšení zaťaženia siete nám môže DiffServ prestať stačiť na rozdiel od IntServ.

IntServ je na druhú stranu veľmi zle škálovateľný. So zväčšovaním siete sa stáva nereálne udržiavať si informácie o všetkých tokoch a ich rezerváciách.

IntServ neponúka všeobecnú metódu ako efektívne prenášať všetky druhy služieb. Použitie IntServ aj na krátkodobé spojenia (napr. http, pop3) je značne neefektívne. Navyše vďaka jejich frekvencii môžu mať veľmi negatívny dopad na výkon routrov. Musíme teda takéto spojenia ignorovať. Tým nám ale môže vzniknúť množstvo premávky v sieti, ktorú nemáme pod kontrolou a nemôžeme tým pádom nič garantovať.

Ďalšia nevýhoda IntServ oproti DiffServ je ťažšia implementácia.

IntServ (RSVP) sa ukazuje ako veľmi vhodný na multimediálne aplikácie v rámci Intranetu.

Kompromisom medzi IntServ a DiffServ je zapojenie obidvoch technológií. Vhodné sa zdá byť takéto zapojenie: IntServ (RSVP) na okrajové siete, kde nie je taká vysoká koncentrácia tokov a nevznikajú tam problémy so škálovateľnosťou a efektivitou. Na chrbticové siete DiffServ – je tam veľa agregovaných tokov. Hraničné routre (DiffServ Boundary Routers) označia RSVP pakety príslušným DSCP. Takisto hraničné routre môžu spájať niekoľko DiffServ domén, potom sa starajú aj o preklad DSCP na základe TCA a SLA. Niekedy nie je iná možnosť ako degradovať alebo povýšiť nejaký paket do inej DSCP.

DiffServ ignoruje RSVP správy (PATH aj RESV) – správa sa k nim ako k normálnym dátam. Ak RSVP router zistí, že od posledného routra zaznačeného v PATH správe prešiel paket cez neRSVP router (napr. porovnaním TTL v IP hlavičke a v tele PATH správy), zaznačí to do ADSPEC dát paketu. Potom už vieme, že nemôžeme poskytnúť garantované služby.

IntServ sa javí ako spojovo-orientovaný (connection-oriented) model, ale funguje nad nespojovanou (conectionless) infraštruktúrou. Jednotlivé implementácie a nasadenia by sa ale mali snažiť zachovávať flexibilitu, ktorú nám ponúka napríklad dynamické routovanie.

2.2.5 MPLS

MPLS nie je technológia, ktorá by sama osebe vedela zabezpečiť QoS. Poskytuje ale prostriedky, ako za pomoci iných technológií tieto vlastnosti zabezpečiť.

Je to vlastne protokol na routovanie, ktorý vytvára spojovo-orientované (connection-oriented) prostredie nad sieťou, ktorá má ne-spojovo-orientované routovanie. V OSI modeli sa zaraďuje medzi druhú a tretiu vrstvu, často označovaný ako 2,5-tá vrstva. Z doposiaľ popísaných je to najnovšia technológia.

Na rozdiel od tradičného prístupu, kde routre smerujú pakety na základe ich cieľovej adresy, MPLS routre smeruje pakety na základe toho, do akej *FEC* (*Forwarding Equivalence Class*) patrí. Zaraďovanie paketov do jednotlivých FEC sa deje na LER routroch pri vstupe paketu do MPLS domény. Priraďovanie sa môže diať nielen na základe dát z druhej vrstvy (IP) ale napríklad podľa portu, alebo routra, z ktorého prišli dáta do siete, teda informácií, ktoré pri bežnom routovaní nie sú k dispozícii.

Príklady možných typov premávky, ktoré môžu byť zoskupené do jednej FEC:

- PQ (Port Quarduplets) – všetky dáta majú rovnaké: sieťová časť zdrojovej a cieľovej adresy, TTL, IP protokol, TCP/UDP zdrojové/cieľové porty
- PQT (Port Quarduplets with TOS) – to isté ako PQ, len navyše majú rovnaké TOS políčko
- HP (Host Pairs) – rovnaké cieľové a zdrojové IP adresy
- NP (Network Pairs) - rovnaké sieťové časti cieľovej a zdrojovej IP adresy
- DN (Destination Network) – rovnaké sieťové časti cieľovej IP adresy
- ER (Egress router) – používané pri OSPF
- NAS (Next-hop AS) – používané pri BGP
- DAS (Destination AS) – používané pri BGP

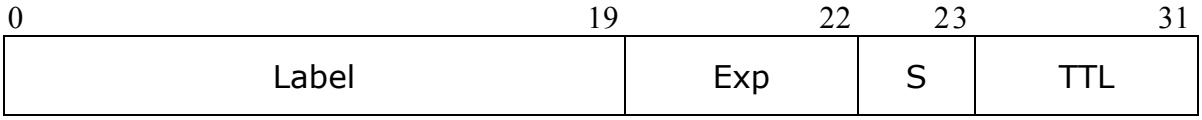
Ďalšou výhodou MPLS oproti bežnému routovaniu je to, že si môžeme dovoliť zložitejšie pravidla bez výrazného výkonového zaťaženia routovania v celej sieti, pretože priraďovanie paketov do jednotlivých FEC sa deje pre každý paket len jeden krát a to pri vstupe do MPLS domény.

Príslušnosť k jednotlivým FEC určuje *label*. Label predstavuje prakticky skrátenú hlavičku paketu. Je pridaný k paketu pri vstupe do MPLS domény.

MPLS doména je svislá oblasť, ktorú tvoria MPLS routre, jej hranica je daná nastavením LER routrov. V súvislosti s MPLS poznáme nasledovné zariadenia a termíny:

- *Label Edge Router (LER)* – sú umiestnené na okrajoch MPLS sietí, priraďujú paketom vchádzajúcim do siete labele.
- *Label Switch Router (LSR)* – smerujú pakety na základe labelu, ktorý v nich nájdu. Môžu zároveň podporovať tradičné metódy smerovania pre neolabelované pakety.
- *Label Switch Paths (LSP)* – trasa pre pakety patriace danej FEC
- *Label Distribution Protocol (LDP)* – protokol na distribuovanie informácií medzi LSR a LER routrami. Exitujúce protokoly ako BG a RSVP boli rozšírené aby akceptovali údaje z LDP.

Každý router priraďuje paketu label, ktorý určuje, kam má byť smerovaný z ďalšieho hopu, teda labele majú len lokálny (pre konrétny router) význam. Keďže mapovanie labelov (zo vstupného na výstupný) je v LSR routroch injektívne, cesta, ktorou bude paket v rámci MPLS domény smerovaný zavisí od prvého labelu, ktorý mu priradí LER router. Štruktúra labelu je na obrázku 4.



Obrázok 4

MPLS teda vlastne simuluje okruhovo-orientovane (circuit-switched) prostredie na paketovo-orientovanom (packet-switched).

V ATM sieťach label môže byť VCI/VPI kombinácia, podobne vo frame relay DLCI. V ostatných sieťach (Ethernet, tokenring, fddi, ppp) musí byť label pridaný medzi hlavičku 2 vrstvy a dáta 3 vrstvy.

2.2.5.1 Hierarchia

MPLS umožňuje agregáciu jednotlivých FEC do iných 'vyšších' FEC, resp. vytváranie hierarchických MPLS domén, ktoré sami obsahujú ďalšie MPLS domény. Toto je umožnené 'zásobníkom labelov'. Teda každý paket môže obsahovať viacero labelov, ku ktorým je pristupované ako k zásobníku. Ak je zásobník prázdny, je paket považovaný za neolabelovaný.

LER routre jednotlivých domén pridávajú pri vstupe paketu do domény na zásobník ďalsi label, LSR routre smerujú pakety na základe tohto – najvyššieho – labela. Pri výstupe z domény sa label zo zasobníka odoberie.

Iné využitie zásobníka labelov je vytváranie virtuálnych tunelov.

2.2.5.2 LDP

Na vytváranie mapovanie medzi FEC a LSP sa používa LDP [RFC LDP] protokol. Tento definuje pravidlá na distribúciu informácií medzi routrami, na oznamovanie (advertising) o prítomnosti jednotlivých routrov v sieti a na správu mapovania FEC na jednotlivé LSP a ich výmenu medzi routrami.

Je dôležité si uvedomiť, že LSR a LER si budujú lokálne mapovania na základe existujúcich routovacích protokolov ako napr OSPF.

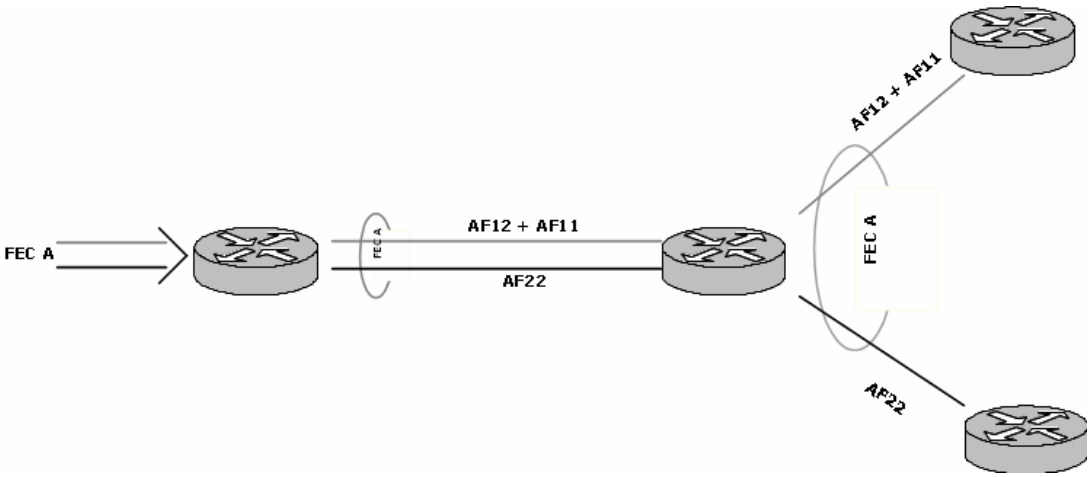
2.2.6 MPLS DiffServ

Na zabezpečenie využitia QoS služieb v MPLS sieťach sa ukazuje vhodné použitie DiffServ. Musíme zabezpečiť konštantné mapovanie z DSCP na PHB v rámci na MPLS routroch. Problém je, že LSR routre smerujú pakety len na základe labelu, teda nemajú prístup do IP hlavičky. Existujú dve možné riešenia:

- využiť EXP políčko v MPLS hlavičke. EXP políčko má ale dĺžku len 3 bity, čo nám nestačí na kompletné mapovanie 8bitového DSCP políčka. Toto riešenie je vhodné pre siete, kde v rámci jednej DiffServ domény máme definovaných maximálne 8 PHB (teda DSCP hodnôt). Smerovanie sa teda deje na základe labelu, navyše sa zabezpečuje PHB správanie na každom MPLS routri. LSP cesta, na ktorej je PHB závislé od EXP bitov, sa nazýva *E-LSP* cesta.

- v prípade sietí, kde vyžadujeme viac ako 8 PHB, musíme využiť iný prístup. Jediná možnosť, ako zabezpečiť všetky potrebné PHB je využiť okrem EXP aj bity labela. Pri inicializácii mapovania FEC na LSP musíme zabezpečiť, aby sa informácii o PHB uložená v labely zachovala v rámci celej domény. LSP cesta, na ktorej je PHB závislé aj od labela, sa nazýva *L-LSP* cesta. L-LSP cesta pre pakety z rovnakej FEC nemusí byť na rozdiel od tradičnej LSP cesty rovnaká.

  Pri tomto riešení sa často používajú EXP bity na uloženie drop preference hodnoty a číslo fronty na prioritizovanie schedulovania sa ukladá do labelu. Teda po jednej L-LSP ceste sa prenášajú pakety, ktoré majú mať zabezpečené PHB AFxy, kde len x je konštantné. Na obrázku 5 je znázornená situácia, kde majú dáta v rámci jednej FEC triedy priradených viacero L-LSP ciest.



Obrázok 5

2.2.7 QoS metódy v Ethernete

QoS služby sa na druhej vrstve zabezpečujú technikami na označovanie paketov:

- 802.1p
- 802.1Q

Vzhľadom na vysokú rýchlosť požadovanú na tejto vrstve, vykonávajú zariadenia len jednoduché funkcie riadení prenosu, spravidla vo forme manažovania jednoduchého algoritmu na fronty.

2.3 Multicasty

Poznáme 4 spôsoby doručovania dát prijímateľom:

1. unicast – dáta sú určené len jednému prijímateľovi
2. broadcast – dáta sú určené všetkým zariadeniam v nejakej oblasti
3. multicast – dáta sú určené niektorým zariadeniam – členom multicastovej skupiny
4. anycast – dáta sú určené jednému zo skupiny. Odosielateľovi nezáleží na tom, ktorému.

Poznáme dva druhy multicastového prenosu

1. LAN multicasting – deje sa na linkovej vrstve
2. IP multicasting – deje sa na sieťovej vrstve

Pre LAN multicasting sú vyhradené fyzické MAC adresy v rozsahu

00:00:5e:00:00:00 – 00:00:5e:7f:ff:ff

Pre IP multicasting sú vyhradené IP adresy v rozsahu

224.0.0.0 – 238.255.255.255

Niektoré multicastové adresy sú vyhradené pre osobitné použitie. Napríklad:

224.0.0.1 – *all-multicast-hosts*
224.0.0.2 – *all-routers*

224.0.0.4 – DVMRP routre

224.0.0.5 – OSPF routre

224.0.0.6 – OSPF routre

Všeobecne rozsahy 224.0.0.0 – 224.0.0.255 a 239.0.0.0 – 239.255.255.255 sú vyhradené na špeciálne účely.

Z uvedeného vidíme, že pre LAN multicastové adresy je vyhradenách 23 bitov, zatiaľ čo pre IP multicastové adresy je vyhradených až 28 bitov. Na preklad IP multicastových adries na MAC adresy sa nepoužíva tradičný ARP protokol, ale neinjektívne mapovanie. Posledných 23 bitov IP adresy sa použije ako 23 bitov v MAC adrese, ktoré sú vyhradené pre multicastové adresy.

Informáciu o umiestnení staníc, ktoré patria do multicastových skupín, nám na sieťovej vrstve zabezpečuje IGMP protokol. Vďaka nemu zariadenia v rámci jedného LAN segmentu vedia o prítomnosti staníc patriacich do skupín. O samotné smerovanie paketov sa potom starajú multicatové routovacie protokoly, ktoré si informácie z IGMP protokolu nejakou formou vedia vymienať.

Iná situácia ale nastáva pri LAN multicastingu. Zariadenia na 2. vrstve nevedia kde sa v LAN segmente nachádzajú členovia multicastových skupín. Pri unicastovom prenose vedia zariadenia druhej vrstvy lokalizovať konkrétne stanice na základe zdrojovej adresy v paketoch, ktoré od nich prichádzajú. Členovia multicastových skupín však neodosielajú pakety so zdrojovou adresou nastavenou na adresu skupiny. Tu sa používa niekoľko techník:

1. broadcast – najjednoduchšie je posielať multicastové dáta ako broadcasty. Máme tým zaručené, že sa dostanú k cieľu.
2. *IGMP snooping* – zariadenia na druhej vrstve rozumejú obsahu IGMP paketov.
3. *CGMP* – protokol vyvinutý firmou Cisco. Pomocou neho sa prepínače dozvedia od routrov informácie o multicastových skupinách.

Na to aby operačný systém podporoval multicasty, musí existovať rozhranie, ktoré môžu aplikácie použiť na prihlásenie alebo odhlásenie sa zo skupiny. Linux podporuje rozhranie pre protokol IGMPv3 od verzie kernelu 2.6.

To, či splňuje nejaká stanica prípadne operačný systém 'multicastovosť' sa určuje tzv. *levels of conformance*. Musia byť definované vždy vzhľadom na konrétnu verziu protokolu IGMP.

Level 0 : žiadna podpora multicastov – multicaastové pakety sa ignorujú
Level 1 : vie posielať multicastové pakety
Level 2 : vie aj prijímať multicastové pakety, prihlasovať sa do skupín (IGMP)

2.3.1 IGMP protokol

Protokol IGMP je protokol určený na prenos informácií o členstve v multicastových skupinách medzi jednotlivými zariadeniami v sieti. Má niekoľko verzií:

2.3.1.1 IGMPv1

Routre vysielajú periodicky *IGMP Host Membership Query* správy na multicastovú adresu 224.0.0.1 (*all-hosts*). Stanica, ktorá je členom nejakej multicastovej skupiny, vyšle po náhodnom časovom intervale odpoved – pre každú skupinu, v ktorej sa nachádza, vyšle jeden *IGMP Host membership report* správu s cieľovou adresou rovnakou, ako je multicastovú adresu danej groupy. Ak stanica dostane *IGMP Host membership report* správu pre multicastovú skupinu, v ktorej sa nachádza, pred vypršaním zvoleného náhodného intervalu, neposiela už *IGMP Host membership report* pre túto skupinu. Nastavenie cieľovej adresy ako adresy multicastovej skupiny zabezpečí, že v ideálnom prípade odpovie na *query* len jedna stanica pre danú skupinu. *Query* aj *report* správy maju hodnotu TTL nastavené na 1, aby sa udržali vrámci jednej LAN siete. Ak sa stanica prihlási do multicastovej skupiny

vyšle automaticky *report*. Toto zabezpečí, že sa router o novom členstve dozvie skorej než vyprší interval medzi posielaním *query*, čo je podľa [RFC IGMP] 125s.

2.3.1.2 IGMPv2

Verzia 2 prináša pre stanice možnosť bezprostredného sa odhlásenia zo skupiny (*IGMP Leave Group* správy) a routrom možnosť vysielať *query* správy pre špecifické skupiny (*IGMP Group-specific Query* správy).

2.3.1.3 IGMPv3

Verzia 3 prináša možnosť vytvárania filtrov, v ktorých stanice určia nielen v ktorých skupinách sa nachádzajú, ale aj od ktorých staníc chcú resp. nechcú prijímať multicastové dáta.

2.3.2 TTL

Políčko TTL v IP hlavičke má pri multicastovom prenose dvojitý význam. Funguje na obmedzenie počtu routrov, cez ktoré paket prejde. Oblasť, ktorú multicastové dáta prejdú sa určuje aj na základe tabuľky 4:

TTL	Rozsah
0	Určené len pre zdrojovú stanicu resp. aplikácie bežiace na nej
1	Určené pre daný LAN segment
<32	Určené pre danú firmu, spoločnosť, organizáciu
<64	Určené pre daný región
<128	Určené pre daný kontinent
<255	Nemá obmedzenú oblasť

Tabuľka 4

Obmedzenie oblasti len na základe TTL políčka sa ukázalo v niektorých prípadoch nedostatočné a boli vyvinuté iniciatívy, aby sa oblasť dala obmedziť aj na základe samotnej multicastovej IP adresy.

2.4 OSPF

V nasledujúcish kapitolách stručne popíšeme fungovanie a základné princípy protokolov OSPF, MOSPF a QOSPF.

OSPF je dynamický IGP link-state routovací protokol. Funguje priamo nad IP vrstvou, IP protocol id 89. Všetky pakety, ktoré používa, by mali mať nastavenú hodnotu TOS políčka na 0, aby boli prioritizované pred ostatnými paketami.

Routre si medzi sebou vymieňajú informácie o kompletnej topológii siete.

Routre môžu medzi sebou vytvárať vzťahy, na základe ktorých si potom vymieňajú informácie.

Uvažujme jednu LAN sieť. Medzi každými dvoma routrami, ktoré majú pripojené k tejto sieti každý aspoň jedno sieťové rozhranie, vzniká *neighborhood* vzťah. Presnejšie povedané *neighborhood* vzťah vzniká medzi sieťovými rozhraniami rotrov nie medzi samotnými routrami. Vo väčšine prípadov stačí používať terminológiu ‘vzťah medzi routrami’, pretože je zrejmé, o ktoré rozhrania sa jedná.

Niektoré *neighborhood* vzťahy môžu prejsť do *adjacency* vzťahov. Iba routre, ktoré sú v *adjacency* vzťahu si vymieňajú informácie o topológii siete.

Point-to-point siet je taká sieť, ktorú tvorí len jednoduchá linka medzi dvama routrami. Nedajú sa k nej pripojiť žiadne ďalšie zariadenia.

Transit siet je taká sieť, ku ktorej je pripojených dva a viacero routrov, a ktorá umožňuje pripojiť okrem týchto dvoch routrov aj iné sieťové zariadenia.

Stub sieť je taká sieť, ku ktorej je pripojený len jeden router, ale umožňuje pripojenie ďalších sieťových zariadení.

Informácie o topológii siete sa vymieňajú v tzv. LSA paketoch. Sú to informácie o routroch, linkách, ich cenách atď. Každý router má vlastnú databázu LSA paketov – *Link State database*. Cieľom je, aby tieto databázy boli pre všetky routre v rámci určitej OSPF oblasti rovnaké.

Informácie o samotných routroch a jejich linkách (sieťových rozhraniach) obsahujú v *Router-LSA* pakety. Informácie o transit sietiach obsahujú *Network-LSA* pakety. Každý Router-LSA resp. Network-LSA obsahuje informácie len o jednom routri resp. sieti. Každý router musí vytvoriť jeden Router-LSA paket s informáciami o sebe.

2.4.1 Priebeh

Uvažujme jeden LAN segment – sieť (súvislá oblasť siete, ktorá je vymedzená zariadeniami 3. vrstvy vykonávajúcimi routovanie. Má priradený jednoznačný identifikátor – adresu siete spolu s maskou siete). Po spustení začne každý router periodicky vysielať *Hello* pakety. Týmito oznamuje ostatným routrom na sieti, že je spustený. *Hello* pakety nie sú routované do iných sietí.

Pokiaľ nami uvažovaná sieť podporuje broadcasty, musí prebehnúť voľba *Designated routera (DR)* a *Backup Designated routera (BDR)*. V prípade, že router j na sieti sám, stáva sa DR a zároveň aj BDR. Ak sú dvaja a viacerý, stáva sa práve jeden z nich DR a práve jeden BDR. DR aj BDR sa volí na základe tzv. *Router priority*, ktorú nastavuje administrátor, prípadne tzv. *Router ID*, čo je jednoznačný identifikátor routra v rámci jedenj OSPF domény. Väčšinou sa zobrazuje podobne ako IP adresy vo forme štyroch čísiel oddelených bodkou. Routre si ho môžu určiť napríklad podľa IP adresy sieťového rozhrania, ktorá je najvyššia.

Úlohami DR routra je:

1. vytvoriť Network-LSA pre danú sieť
2. vytvoriť si adjacency vzťah so všetkými routrami v danej sieti

BDR musí splniť len bod 2. Týmto je zaručené, že pre každú transit sieť bude existovať práve jeden Network-LSA paket.

BDR sa stane DR routrom, ak DR router spadne.

Ďalším krokom je výmena *Database description* paketov. Tieto pakety obsahujú zoznam LSA paketov, ktoré obsahuje router vo svojej link state databáze. Každý LSA paket sa dá jednoznačne určiť podľa *Link State ID*, teda routre po výmene

Database description paketov vedia, o ktoré pakety majú svojho adjacency suseda požiadať.

Routre žiadajú o LSA pakety, ktoré nemajú vo vlastnej link state databáze. Následne o ne žiadajú pomocou *LSA-Request* paketov. Po prijatí požadovaného *LSA-Update* paketu, potvrdia prijatie pomocou *LSA-Ack* paketu.

Po pridaní nového LSA paketu do databáze, routre vysielajú *LSA-Update* aj do iných sietí, ku ktorým sú pripojené – *flooding procedure*.

2.4.2 Domény, oblasti

OSPF doména – súvislá oblasť siete , v ktorej prebieha OSPF routovanie. OSPF umožňuje administrátorom rozdeliť jednu OSPF doménu na niekoľko oblastí *(OSPF area)*. Každá oblasť má jednoznačný identifikátor *Area ID*. Každá doména musí obsahovať minimálne oblasť s Area ID 0. Táto sa nazýva chrbticová (*backbone*) oblasť. Každá iná oblasť v doméne musí byť priamo pripojená s chrbticovou oblasťou. T.j. pre každú oblasť inú ako chrbticovú musí existovať router, ktorý obsahuje sieťové rozhranie pripojené k chrbticovej sieti a rozhranie pripojené k danej oblasti. Takýto router sa nazýva *Area border router*.

Protokol OSPF sa snaží udržať konzistentnosť link state databází v rámci OSPF oblasti. Všetky dáta, ktoré smerujú z jednej oblasti do inej, musia ísť cez chrbticovú oblasť.

Area border routre posielajú tzv. *Area-Summary-LSA* routrom v chrbticovej sieti. Tieto LSA pakety obsahujú agregované informácie o linkách a ich cenách v rámci jednej oblasti. Routre v chrbticovej oblasti majú teda informácie o všetkých llinkách vo všetkých oblastiach. Cez chrbticovú sieť si teda vymieňajú area-summary-LSA pakety aj samotné area border routre medzi sebou. Každý area border router má teda informácie o linkách v každej oblasti a navyše pozná topológiu chrbticovej oblasti. Takže pokiaľ dostane paket, ktorý má byť smerovaný sez chrbticovú sieť do inej oblasti, vie si spočítať cenu cesty v chrbticovej oblasti a cenu cesty v cieľovej oblasti a teda nájsť najideálnejšiu cestu.

2.4.3 SPF algoritmus

Cieľom OSPF protokolu je naplniť routovaciu tabuľku. Routre musia vedieť pre každý paket výstupne sieťové rozhranie prípadne router (*next hop*), ktorému má dáta ďalej poslať. Routre hľadanú tieto údaje tak, aby cesta, ktorú paket prejde mala čo najnižšiu cenu. OSPF protokol na to používa SPF (shortest path first) algoritmus, ktorý vychádza z Djikstrovho algoritmu.

Cieľom je vytvoriť strom pokrývajúci graf, ktorý znázorňuje oblasť, ku ktorej je router pripojený. Ak je router pripojený k viacero oblastiam, vytvára strom pre každú takúto oblasť.

V *link-state grafe* vytvorenom na základe link state databázi pre danú oblasť predstavujú vrcholy routre a transit siete. Ak obsahuje router rozhranie pripojené k niektorej transit sieti, existuje hrana medzi týmto routrom a transit sieťou. Ak sú dva routre prrpojené point-to-point sieťou, existuje hrana medzi nimi. Stub siete nie sú reprezentované vrcholmi. Hrany sú obojsmerné. Každej hrane je priradená cena, ktorá sa zistí na základe údajov na prislúchajúcom sieťovom rozhraní, z ktorého hrana smeruje.

Algoritmus hľadajúci strom na takomto grafe prebieha v dvoch fázach:

1. Majme zoznam kandidátov (*candidate list*), vytváraný SPF strom a zoznam vrcholov, ktoré reprezentujú routre alebo transit siete (sú to vlastne Router-LSA a Network-LSA pakety z link-state databázy). Na začiatku sú zoznam kandidátov aj SPF strom prázdne.V zozname kandidátov si budeme udržovať vrcholy, ku ktorým sa už našla cesta, ale nie je isté, že táto cesta je najkratšia možná. Vykonávame takýto algoritnus:

    1. Do SPF stromu dáme vrchol predstavujúci samého seba - router, ktorý vykonáva výpočet. Tento bude predstavovať koreň stromu.
    2. Označme vrchol, ktorý sme práve pridali do SPF stromu vrchol V. Prechádzame všetky vrcholy W, ktoré s ním susedia v link-state grafe. Vypočítame novú cenu pre cestu do vrchola W ako súčet ceny cesty do vrchola V a linky z vrchola V do vrchola W.

Ak ešte nemáme záznam o ceste do vrchola W (nenachádza sa v zozname kandidátov ani v SPF strome), uložíme ho do zoznamu kandidátov.

Ak už máme záznam o ceste do vrchola W a nová vypočítaná cesta je drahšia, prejdeme na ďalší vrchol W.

Ak je nová cesta lacnejšia, nahradíme údaj v zozname kandidátov.

Takto prejdeme všetky vrcholy W.

3. Ak je zoznam kandidátov prázdny, skončili sme. Inak vyber zo zoznamu kandidátov vrchol, ktorý je najbližšie ku koreňu (poznáme cestu s najmenšou cenou) a pridaj ho do SPF stromu. Modifikujeme routovaciu tabuľku pre daný vrchol.

2. Prechádzame všetky routre a ich linky, ktoré vedú do stub sietí. Spočítame cenu cesty pre ne na základe vytvoreného SPF stromu. Do routovacej tabuľky vložíme najlacnejšiu akú nájdeme.

3. Podobne sa určia ceny do iných oblastí, prípadne mimo OSPF domény.

Výstupom algoritmu je routovacia tabuľka, ktorá obsahuje pre každú podsieť v OPSF doméne informáciu o tom, ako smerovať pakety určené do tejto podsiete. Táto informácia obsahuje dvojice

- *outifs* - výstupné sieťové rozhranie, ktorým sa má paket poslať
- *nexthop* - adresa ďalšieho routra, ktorému sa má paket poslať

Informácia o next hop chýba v prípade, že paket smeruje do priamo pripojenej siete.

Ak routovacia tabuľka obsahuje pre niektorú cieľovú podsieť takých dvojíc viac ako jednu, môže router posielať dáta

2.4.4 QOSPF

Nerieši nehomogénne prostredie s čistoOSPF routrami. Algoritmus berie do úvahy len prenosovú šírku a snaží sa minimalizovat počet hopov. Potreba prípadného obmedzenia latencie môže byť implementovaná vytvorením určitej "*policy*", ktorá z grafu siete siete vyhodí všetky linky, ktoré nespíňajú určité podmienky (napr. majú väčšiu latenciu, ako je požadovaná).

QOSPF predpočítava QoS cesty podobne ako OSPF. Predpočítanie počíta cesty pre všetky možné požiadavky. Predpočítanie sa môže diať buď periodicky alebo po prijatí určitého počtu LSA-updatov.

QoS parametre liniek sa uchovávajú v *extended TOS* políčku, ktoré sa nachádza v Router-LSA pre každú jeho linku. Predávajú sa prenosová šírka linky a latencia. Updaty (LSA) QoS parametrov sa nemôžu posielat pri každej zmene. Takisto nie je dobré riešenie posielať ich v nejakých pravidelných intervaloch. Možné riešenie je posielať updaty len pri vyrazných zmenách v QoS parametroch (zmeny môžu byť merané buď absolútne alebo relatívne – v %). Popri tom ešte môže fungovať aj periodické posielanie updatov. Metriky posielané v updatoch môžu byť buď presné hodnoty, alebo kvantizované.

Kódovanie QoS parametrov do TOS policko sa snaží sa byť do určitej miery spätne kompatibilné s [RFC TOS]. TOS je 5 bitové. Metrika má priestor 16bitov. Pri kódovani QoS hodnôt kódujeme exponenciálne – ako mantisu a exponent. Kódovanie je zvolené tak, aby routre bez podpory QoS, ktoré berú metriku ako cenu (cost) interpreovali toto políčko relatívne spravne (ked je malá prenosová šírka, tak je veľká cena a opačne). Zaokrúhlovanie skutočnej hodnoty na najbližšiu robíme pesimistickým prístupom (pri prenosovej šírke nadol, pri latencii nahor).

2.4.5 MOSPF

MOSPF je multicastové rozšírenie protokolu OSPF definované v [RFC MOSPF]. Umožňuje routovanie multicastových paketov.

MOSPF routre používajú protokol IGMP na získavanie informácií o členstve staníc, ktoré sa nachádzajú v priamo pripojených sietiach, v multicastových skupinách. Získané informácie si udržujú v *lokálnej multicastovej databáze*. Tá obsahuje dvojice:

- multicastová skupina resp. jej adresa
- adresa priamo pripojenej siete

Informácie z tejto tabuľky sa prenášajú k ostatným routrom pomocou *Group-Membership-LSA* paketov. Pre každú multicastovú skupinu, o ktorej existuje informácia v lokálnej databáze sa vygeneruje jeden Group-Membersihp-LSA paket.

Pre transit siete generuje Group-Membership-LSA pakety Designated router.

2.4.6 Multicastové routovanie

Samotné routovanie sa v MOSPF deje nielen na základe cieľovej adresy paketu, ale aj na základe zdrojovej adresy.

MOSPF routre nemajú predvypočítanú smerovaciu tabuľku pre všetky možné kombinácie zdrojových a cieľových adries. Toto by bolo neúnosne náročné nie len na výpočet, ale aj na následné uloženie takejto smerovacej tabuľky. Namiesto toho vypočítavajú MOSPF routre SPF stromy ‘na požiadanie’ (*on demand*).

MOSPF routre nepodporujú multipath smerovanie. Pre každú dvojicu zdrojovej a cieľovej adresy existuje len jedna možnosť ďaľšieho smerovania.

MOSPF vytvára SPF strom, ktorý nemá koreň v routri, ktorý vykonáva výpočet, ale koreň stromu sa nachádza vo vrchole predstavujúcom zdroj paketu. Teda všetky routre, cez ktoré je smerovaný určitý paket, vytvárajú rovnaký SPF strom.

Na výpočet SPF stromu sa používa mierne upravený algoritmus z OSPF protokolu. Hlavná zmena je v pridávaní vrchola do vytváraného SPF stromu. Po pridaní musí MOSPF router navyše zistiť, či neobsahuje pridaný vrchol členov cieľovej multicastovej skupiny. V prípade, že ano, nájdeme vo vytváranom strome náš router a pridáme do zoznamu výstupných sieťových rozhraní pre danú skupinu a zdrojovú adresu rozhranie, ktorým sa v strome dostaneme od nášho routra ku pridávanému vrcholu. Ak máme nájdený v strome nás router, vieme si zistiť aj vstupné

sieťové rozhranie, ktorým musel byť paket prijatý. Ak sme ho prijali iným rozhraním, než aké sme zistili z SPF stromu, nastala pravdepodobne situácia, kedy je LAN multicasting riešený formou broadcastov.

Algoritmus na výpočet multicastového stromu nájde najoptimálnejšiu (najlacnejšiu) cestu pre danú kombináciu zdrojovej a cieľovej adresy, teda najoptimálnejšiu vzhľadom na jednotlivé cieľové stanice. Nehľadá však multicastový strom najoptimálnejší vzhľadom na celkové zaťaženie siete.

Ak sa v OSPF doméne nachádzajú okrem MOSPF routrov aj OSPF routre bez podpory multicastov, takéto routre sa pri výpočte multicastového stromu v MOSPF routroch nepoužívajú. Naopak OSPF routre ignorujú Group-membership-LSA pakety. Problém môže vzniknúť vtedy, keď sa multicastová sieť stane nespojitou, prípadne ak je na niektorej sieti zvolený DR router nepodporujúci MOSPF. Routre svoje multicastové rozšírenie oznamujú v Hello paketoch.

3 Použitie – zhrnutie

Teoretické znalosti v oblasti multicastov a QoS sú dostatočné na to, aby tieto dve technológie mohli výrazne pomôcť prekonať problém čoraz náročnejších internetových aplikácií. Prax však ukazuje niečo iné.

3.1 QoS

Väčšina nami popísaných technológií bola vyvinutá v 90. rokoch minulého storočia. V tej dobe prežívalo QoS veľký rozmach a v ústach odborníkov (ale aj manažérov) to bolo magické slovíčko, ktoré bolo riešením všetkých problémov.

Postupom času sa ale situácia obrátila. Ideálna predstava siete, ktorej keď zadáme požiadavku na (multicastové) spojenie so zadanými QoS parametrami, sieť odpovie buď kladne alebo záporne a v prípade kladnej odpovede nám poskytne spojenie o ktorom si môžeme byť istý, že po celú dobu trvania si zachová nami požadované vlastnosti, sa ukázala nereálna. Takúto predstavu spočiatku spĺňali služby IntServ, neskôr sa ukázalo, že z dôvodu škálovateľnosti bude lepšie použitie IntServ v kombinácii s DiffServ službami. Už pri tejto kombinácii však musíme v našej predstavy urobiť určité ústupky. Sieť nám pri použití DiffServ služieb nemôže nič zaručiť. Ostáva nám spoliehať sa na to, že sieť je dobre navrhnutá, a že DiffServ služby majú vyhradenú prenosovú šírku dosť veľkú na prenos dát, ku Ktorým vyžadujeme rezerváciu zdrojov.

Tento ústupok sa ukázal kritický, pretože prakticky rušil funkčnosť a význam služieb IntServ. Toto si uvedomili administrátori a takisto aj výrobcovia sieťových zariadení.

Implementácia služieb IntServ je náročnejšia ako služieb DiffServ. Pri službách DiffServ ide prakticky len o prioritizáciu, zatiaľ čo pri službách IntServ musíme implementovať protokol pre rezerváciu zdrojov, zariadenie musí byť schopné udržovať si informácie o prebiehajúcich tokoch, zložité a stále diskutabilné je správanie sa pri zmene v zaťažení siete atď.

Vďaka týmto skutočnostiam sa na trhu sa zariadenia podporujúce služby IntServ prakticky nikdy neobjavili. Problém nebol (a nie je) len so sieťovými zariadeniami ale aj s koncovými stanicami.

Ďalší problém globálneho nasadenia QoS je QoS routovanie medzi AS. Tu už sa jedná o právne problémy. Spoločnosti, ktoré chcú používať QoS, uzatvárajú zmluvy so svojimi ISP (Internet Service Provider). Väčšinou je treba dohodnúť sa, ako budú klasifikované dáta pri vstupe do siete ISP. Toto je obsahom TCA (Traffic Conditioning Agreement). Obsahom SLA (Service Level Agreement) sú charakteristiky prenosu, ktoré budú poskytnuté jednotlivým službám (definovaným v TCA).

Tieto zmluvy sú vždy kompromisom medzi ISP a zákazníkom a ich uzatváranie je často zdĺhavý a finančne náročný proces, ktorý nikdy stopercentne neuspokojí potreby zákazníka. Navyše tieto zmluvy nehovoria nič o tom, ako sa budú dáta prenášať mimo sieť ISP. Bolo by teda treba, aby ISP uzatvárali podobné zmluvy aj medzi sebou. Vzhľadom k neustálej kulminácii zákazníkov a zmenám ich požiadaviek by to ale bol nekonečný právno-obchodný proces.

Tieto a ďalej uvedené problémy boli príčinou toho, že použitie QoS sa zúžilo len na použitie základných princípov služieb DiffServ – prioritizácie implementovanej len v rámci jednotlivých ISP a ich poskytovanie na základe jednoduchých zmlúv so zákazníkmi. Prioritizáciu si dosť často ISP implementujú ‘po svojom’, vytvoria si vlastné priradenie TOS hodnôt službám, niekedy aj vlastné mechanizmy na správu front. Toto pramení hlavne z neznalosti QoS techník. V lepšom prípade použijú sieťové zariadenie prípadne softvér na routovanie, ktorý má implementované služby DiffServ.

V poslednom čase sa začína presadzovať technológia MPLS. Je ideálna na použitie spolu s DiffServ. Niektoré princípy ich fungovania sú dokonca podobné: obidve technológie pri vstupe paketu do domény paket niečím označia, na základe čoho je potom v prípade MPLS smerovaný a v prípade DiffServ schedulovaný. Ďalšia nová technológia alebo skôr termín je Traffic Engineering (TE). Jedná sa o služby poskytujúce garanciu QoS nárokov. Používa sa v súvislosti s predchádzajúcimi službami, čím vznikajú názvy ako napr. „MPLS-TE DiffServ“. Tieto pojmy majú skôr marketingový charakter. Spomínané technológie sú dosť nové a nie sú ešte ani presne špecifikované, resp. si ich jednotlivý výrobcovia definujú po svojom.

3.2 Multicasty

Problém globálneho nasadenia multicastov je škálovateľnosť. S narastajúcim počtom AS a mlticastových skupín vzniká veľký problém ako si udržať, vymienať a zistiť informácie o príslušnosti jednotlivých staníc ku skupinám.

Bolo vyvinutých niekoľko protokolov (DVMRP) a techník, ktoré sú založené nie na pripájaní sa staníc ku skupinám, ale na odmietaní dát prislúchajúcich nechcenej skupine. Vychádzajú z toho, že sieť má kapacitu na prenos multicastových dát tradičným neefektívnym unicastovým spôsobom resp. až broadcastovým spôsobom. Nesnaží sa za každú cenu dodržať zásady multicastového prenosu. Prioritou je zaistenie prenosu dát, efektivita je až na druhom mieste.

Ďalším problémom multicastov je zaručenie homogénneho multicastového prostredia. Výskyt jediného ne-multicastového zariadenia môže úplne narušiť fungovanie multicastových prenosov. Riešenie je vo vytváraní tunelov cez štandardné unicastové zariadenia. Takto vznikla napríklad sieť virtuálna sieť MBone, ktorá pracuje nad existujúcim Internetom. Do tejto siete sa môže pripojiť ktokoľvek s potrebným technickým vybavením. Prebiehajú na nej napríklad priame prenosy z vedeckých konferencií.

3.3 Prenosová šírka zadarmo

Jedným z najčastejším argumentom proti QoS a multicastovým technológiám je: „Prenosová šírka je *lacnejšia* ako implementácia QoS a multicastových technológií a do budúcna sa bude ešte zlacňovať.“

Vzhľadom k už popísanej prenosovej náročnosti dnešných aplikácií a pohľadu na dnešný stav Internetu sa ale domnievame, že v niektorých častiach sveta bude toto tvrdenie ešte pár rokov neplatné. Argument o prenosovej šírke zadarmo väčšinou používajú zamestnanci obrovských amerických spoločností. V jejich prípade majú možno pravdu.

My sa domnievame, že toto tvrdenie nemôžeme použiť všeobecne, jeho pravdivosť závisí od konkrétnej situácie. Použitie jednoduchej dvojúrovňovej

prioritizácie medzi ISP a jej zákazníkom môže priniesť pre zákazníka výhody v podobe spoľahlivejšieho prenosu kritických dát. Multicastové prenosy môžu pomôcť v prípade, že v sieti zákazníka sa nachádza viacero (nie všetci) účastníkov multicastového prenosu. Vytvorenie jednoduchého tunela medzi zdrojovou stanicou a cieľovou sieťou nie je náročný proces, ale môže výrazne znížiť jednorazové zaťaženie pripojenia zákazníka ku ISP.

Ďalším dôvodom proti tomuto tvrdeniu je, že pri zvyšovaní kapacity sa používaných liniek a jej zlacňovaní sa automaticky objavja ďalšie náročnejšie aplikácie, ktoré túto prenosovú šírku pohltia. Príkladom môže byť príchod 3G sietí, pri ktorých automaticky vznikla možnosť internetového televízneho vysielania, videohovorov, sťahovania filmov atď. Tieto siete využívajú QoS princípy v určitej podobe.

Potrvá ešte nejakú dobu, kým bude prenosová kapacita tak lacná, že ňou bude možné plytvať. Do tej doby majú QoS a multicaty uplatnenie.

3.4 IPv6

Protokol IPv6, ktorý je prirodzený nástupca dnes používaného protokolu IPv4, implicitne podporuje QoS aj multicastové techniky.

Taktiež môžeme tvrdiť, že do doby masového rozšírenia protokolu IPv6 budú mať štandardné QoS a multicastové techniky pre IPv4 uplatnenie.

3.5 Stupid nets

Niektorý odporcovia QoS a multicastov sú toho názoru [StupidNets], že sieť by mala mať čo najmenšiu inteligenciu a jediné na čo by sa mala zamerať je, aby nejakým spôsobom preniesla dáta tam, kam majú byť prenesené.

Táto teória sa nám zdá veľmi „extrémna“. David Isenberg, autor tejto teórie, zjednodušuje situáciu okolo počítačových sietí.

4 Praktická časť

Cieľom praktickej časti bolo vytvoriť riešenie pre OSPF routovanie s použitím QoS a multicastových mechanizmov.

Na základe úvah z predchádzajúcej kapitoly sme sa rozhodli pre implementáciu multicastového rozšírenia protokolu OSPF s použitím služieb DiffServ.

Pri implementácii multicastového rozšírenia sme vychádzali zo štandardu MOSPF [RFC MOSPF].

Ako nástroj na zabezpečenie QoS služieb sme vybrali tcng, ktorý umožňuje širokú konfiguráciu front a QoS tried.

Pri implementácii multicastového rozšírenia sme vychádzali z routovacieho balíka Quagga. Je to jedna z najrozšírenejších voľne šíriteľných implementácií protokolu OSPF. Quagga funguje pod operačnými systémami Linux. My sme zvolili distribúciu Debian GNU/Linux 3.1r4 (Sarge). Je to stabilná (*stable*) vetva Debianu. Používa verziu kernelu 2.4.27. Všetky testy sme robili na platforme i386.

Na koniec kapitoly uvedieme jednoduchý test, ktorý ukáže možné využitie QoS a multicatov v praxi a preverí základnú funkcionalitu našej implementácie.

4.1 Quagga

Ako základ fungujúceho OSPF protokolu sme použili routovací balík Quagga (www.quagga.net). Vychádzali sme z verzie 0.98.6. Quagga je voľne šíriteľná na základe licencie GNU GPL. Zdrojové kódy sú v jazyku C.

Quagga používa architektúru modulov. Každý modul je spustený ako samostatný démon. Základný modul Quaggy je Zebra. Tento poskytuje ostatným modulom informácie hlavne ohľadom sieťových rozhraní routra a routovacej tabuľky. Jednotlivé moduly môžu byť spustené na rozličných počítačoch.

Každý routovací protokol, ktorý Quagga podporuje, je implementovaný v samostatnom module. V súčastnosti má Quagga moduly pre RIP, OSPF, BGP, IS-IS. Na komunikáciu s modulom Zebra sa využíva tzv. Zebra protokol. Ten umožňuje

prácu s routovacou tabuľkou a sieťovými rozhraniami. Našu implementáciu MOSPF sme urobili ako modifikáciu modulu OSPF, konkrétne pre OSPFv2.

Konfigurácia Quaggy prebieha podobne ako väčšiny Cisco zariadení. Každý modul má vlastnú konfiguračnú konzolu. V module Zebra sa konfigurujú sieťové rozhrania a statické routovanie.

Pri implementácii sme využívali existujúce knižničné nástroje Quaggy na obsluhu logovania, práce s vláknami, schedulovanie vlákien, ktoré čakajú na dáta zo socketa atď.

Vzhľadom na náročnosť naša implementácia nepodporuje multicastové prenosy medzi OSPF oblasťami ani jednotlivými AS.

4.2 Implementácia MOSPF

Do Quaggy sme pridali nasledovné zdrojové súbory:

1. ospfd/ospf_mospf.c, ospfd/ospf_mospf.h – vychádzajú zo súboruov ospf_spf.*. Obsahuje implementáciu výpočtu multicastového stromu, modifikovania multicastovej routovacej tabuľky.
2. ospfd/ospf_igmp.c, ospfd/ospf_igmp.h – obsahujú implementáciu serverovej časti protokolu IGMP, zabezpečujú obnovovanie databázy Group-membership-LSA záznamov.

Modifikované boli tieto súbory:

1. ospfd/ospfd.c, ospfd/ospfd.h – pridaná tabuľka mospf_inst_mc_routes do štruktúry ospf, jej inicializácia
2. osfpd/ospf_lsa.c, osfpd/ospf_lsa.h – pridaný typ Group-membership LSA, funkcie na prácu s ním
3. osfpd/ospf_lsds.c – vytvorená databáza pre Group-membership LSA
4. ospfd/ospf_vty.c – pridané nové príkazy pre prácu s lokálnou multicastovou tabuľkou
5. lib/memory.h – pridané funkcie na memory management nových štruktúr

Implementácia multicastového rozšírenia pozostávala z niekoľkých krokov:

1. implementovať protokol IGMP a tabuľku s dátami, ktoré z protokolu získame - príslušnosť priamo pripojených sietí do multicastových skupín
2. zebezpečiť výmenu získaných informácií medzi routrami pomocou Group-Membership LSA paketov
3. detekcia prichádzajúcich multicastových dát a následné vytvorenie konkrétneho SPF stromu
4. pridanie cesty pre multicastové dáta do routovacej tabuľky linuxového kernelu

4.2.1 IGMP

Rozhodli sme sa implementovať IGMPv2. Cieľom bolo teda implementovať routrovskú časť protokolu. Jadro Linuxu 2.4.27 obsahuje klientskú časť implementácie všetkých verzií IGMP.

Naša verzia dokáže čítať pakety IGMPv3, ktoré sú spätne kompatibilné s IGMPv3 napr. *Host membership report*.

Do štruktúry *ospf* sme pridali zoznam *local_mc_group_db*. Ten implementuje tabuľku, ktorej záznamy majú štruktúru:

```
struct local_mc_group_db_entry {
        struct ospf *ospf;
        struct in_addr mc_group;
        struct list * ointerfaces;
        struct ospf_lsa *group_member_lsa_self;
        struct thread *t_group_member_lsa_self;
};
```

Každý záznam má jedinečný prvok *mc_group*, v ktorom je uložená multicastová adresa. V prvku *ointerfaces* je zoznam rozhraní, na ktorých bola prijatá IGMP správa o členstve v skupine mc_group.

Pri inicializácii OSPF routovania sa otvárajú nové sockety: *ospf_igmp_recv_socket* a *ospf_igmp_send_socket*. Sú to sockety pre triedu protokolov

PF_PACKET. Táto trieda je dostupná od verzie kernelu 2.2. Umožňuje odchytávanie paketov na úrovni sieťovej IP vrstvy, kde vieme rozoznať, či sa jedná o paket protokolu IGMP.

Recv socket je otvorený pre všetky sieťové rozhrania a zachytáva všetku komunikáciu, nielen tú, ktorá je určená tomuto routru (*promiscuous mode*). Tým sme zabezpečili, aby videl aj IGMP pakety určené skupinám, v ktorých sa samotný router nenachádza. Soket má nastavený *BPF* filter na akceptovanie paketov smerujúcich len na multicastové adresy. BPF filter pracuje na princípe konečného automatu. Pracuje na nízkej úrovni vďaka čomu je veľmi rýchly. Recv socket testuje, či sa jedná o dáta protokolu IGMP (funkcia *ospf_igmp_recv_data*), ak áno, spracuje ich obsah a prípadne pridá alebo upraví záznam v tabuľke *local_mc_group_db.*

Send socket je používaný na vysielanie IGMP query paketov cez sieťové rozhrania, ktoré sú členmi OSPF domény.

4.2.2 Group-Membersihp-LSA

Formát paketov Group-Membership-LSA (gmLSA) sme prevzali z [RFC MOSPF] dokumentu:

```
struct group_member_lsa
{
  struct lsa_header header;
  struct {
        uint32_t vertex_type;
#define GROUP_MEMBER_LSA_VERTEX_ROUTER              1
#define GROUP_MEMBER_LSA_VERTEX_NETWORK             2
        struct in_addr vertex_id;
  } vertex[1];
};
```

Pri implementácii funkcií na správu a výmenu týchto paketov sme vychádzali hlavne z existujúcich funkcií pre Network-LSA pakety. Vzťah sieťové rozhranie – Network-LSA je v mnohom podobný ako vzťah multicastová skupina – Group-Membership-LSA.

Na výmenu, správu, obnovovanie a vytváranie nových gmLSA bol využitý existujúci framework, založený na knižničnej funkcionalite vlákien a socketov. Obsluha gmLSA sa deje pomocou nasledovných funkcií:

- *ospf_group_member_lsa_timer_add* – pridá timer, ktorý zavolá funkciu ospf_group_member_lsa_timer_add po uplynutí štandardného (z nastavenia pre všetky typy LSA) času
- *ospf_group_member_lsa_refresh_timer* – zavolá fukciu pre update prípadne vytvorenie nového gmLSA
- *ospf_group_member_lsa_originate* – vytvorí nový gmLSA, pridá do LSAdb
- *ospf_group_member_lsa_body_set* – vytvorí telo gmLSA – štruktúru podľa [RFC MOSPF]
- *ospf_group_member_lsa_new* – vytvorí nový gmLSA. Nastaví mu atribúty potrebné na jeho obnovovanie, distribúciu a schedulovanie týchto činností
- *ospf_group_member_lsa_install* – pridá gmLSA do LSAdb

4.2.3 Multicastový SPF strom

Na detekciu prichádzajúcich paketov sa používa *ospf_igmp_recv_socket*. Pakety sú z neho smerované buď do IGMP rozhrania alebo sú brané ako multicastové dáta, ktoré je treba routovať.

Dáta, pre ktoré existuje záznam v multicastovej routovacej tabuľke, si udržujeme v zozname *mospf_inst_mc_routes* (cache pamäti) so záznamami:

```
struct mospf_mc_route
{
        struct in_addr srcaddr;
        struct in_addr mc_group;
};
```

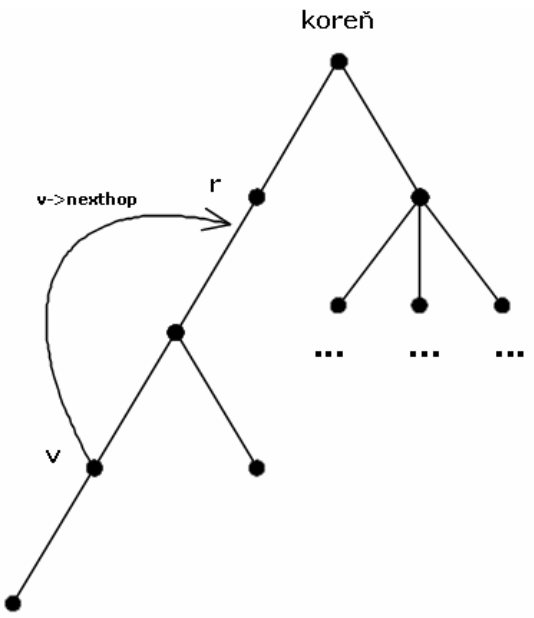
Ak sa v cache pamäti nenáchádza cesta pre daný multicastový paket, musí sa vypočítať. Deje sa tak vo funkcii *ospf_mospf_spf_calculate*. Funkcia nájde rozhrania, cez ktoré by náš router mal poslať tento paket a pakety s rovnakou zdrojovou a cieľovou adresou.

Funkcia vypočítava SPF strom pre danú kombináciu zdrojovej adresy a multicastovej skupiny, pre ktorú je paket určený. Vrcholy stroma majú štruktúru:

```
struct mospf_vertex
{
  u_char flags;
  u_char type;         /* copied from LSA header */
  struct in_addr id;   /* copied from LSA header */
  struct lsa_header *lsa; /* Router or Network LSA */
  u_int32_t distance;  /* from root to this vertex */
  struct list *child;          /* list of vertex: children in SPF
tree*/
  struct mospf_vertex_nexthop *nexthop;              /*
vertex_nexthop from this router to this vertex */
  struct mospf_vertex *parent; /* parent in SPF tree */
};
```

Prvok *nexthop* obsahuje rozhranie, ktorý je na ceste od koreňa smerom od vrcholu, ktorý predstavuje náš router (na ktorom prebieha výpočet). Má štruktúru:

```
struct mospf_vertex_nexthop
{
  struct ospf_interface *oi;   /* output intf on root node */
  struct in_addr router;       /* router address to send to */
};
```



Obrázok 6

Na obrázku 6 je znázornený prvok nexthop. Na obrázku je SPF strom, kde vrchol *r* predstavuje router, na ktorom prebieha výpočet.

Výpočet pre paket so zdrojovou adresou *srcaddr* a cieľovou *mc_group* prebieha v nasledujúcich krokoch:

1. Na začiatku treba vytvoriť koreň stromu a nájsť LSA (*ospf_mospf_find_lsa)*, ktoré bude znázorňovať. Toto LSA môže byť typu
   - Network LSA - ak zdrojová adresa patrí do niektorej tranzitnej siete
   - Router LSA – ak zdrojová adresa patrí do stub siete, niektorého routra
2. Pokračujeme cyklom, ktorý sme opísali v kapitole *MOSPF*, resp. ktorý je definovaný v [RFC MOSPF]. Počas cyklu si udržujeme zoznam rozhraní *outifs*, kde sú výstupné rozhrania, ktoré majú byť výsledkom výpočtu. Pri pridávaní vrcholu do stromu skontrolujeme, či sa v sieti predstavujúcej vrchol nenachádzajú členovia *mc_group*. Ak áno, pridáme rozhrania z *nexthop* do zoznamu *outifs*.
3. Pri prechádzaní stub sietí hľadáme také, ktoré obsahujú členov *mc_group*. Do zoznamu outifs potom pridávame rozhranie z nexthop štruktúry.
4. Nakoniec pridáme nový záznam do multicastovej routovacej tabuľky pomocou nástroja smcroute.

4.2.4 Obsluha multicastovej routovacej tabuľky

Na manipuláciu so záznamami z multicastovej routovacej tabuľky linuxového kernelu sme použili existujúci nástroj *smcroute* (www.cschill.de/smcroute). Cache pamäť pre multicastové routovanie v linuxovom kernely pozostáva zo štvoríc:

```
[srcaddr, mc_group, inif, outifs]
```

, kde:

- srcaddr - zdrojová adresa
- mc_group - cieľová multicastová adresa
- inif - vstupné rozhranie, ktorým bol paket prijatý

- outifs - zoznam výstupných rozhraní, ktorými sa má paket poslať ďalej

Všetky tieto údaje vieme zistiť zo skonštruovaného SPF stromu. Pridávanie záznamov sa deje vo funkcii *ospf_mospf_add_mc_route*.

4.2.5 Modifikácia konzoly

V nasledujúcich kapitolách budú popísané príkayz, ktoré sme pridali do konfiguračnej konzoly pre modul ospf.

4.2.5.1 show ip ospf local-group-database

Vypíše obsah tabuľky *local_mc_group_db*, teda multicatových skupín, ktorých členovia sú v niektorej zo sietí pripojených na rozhranie routra. Príklad výstupu:

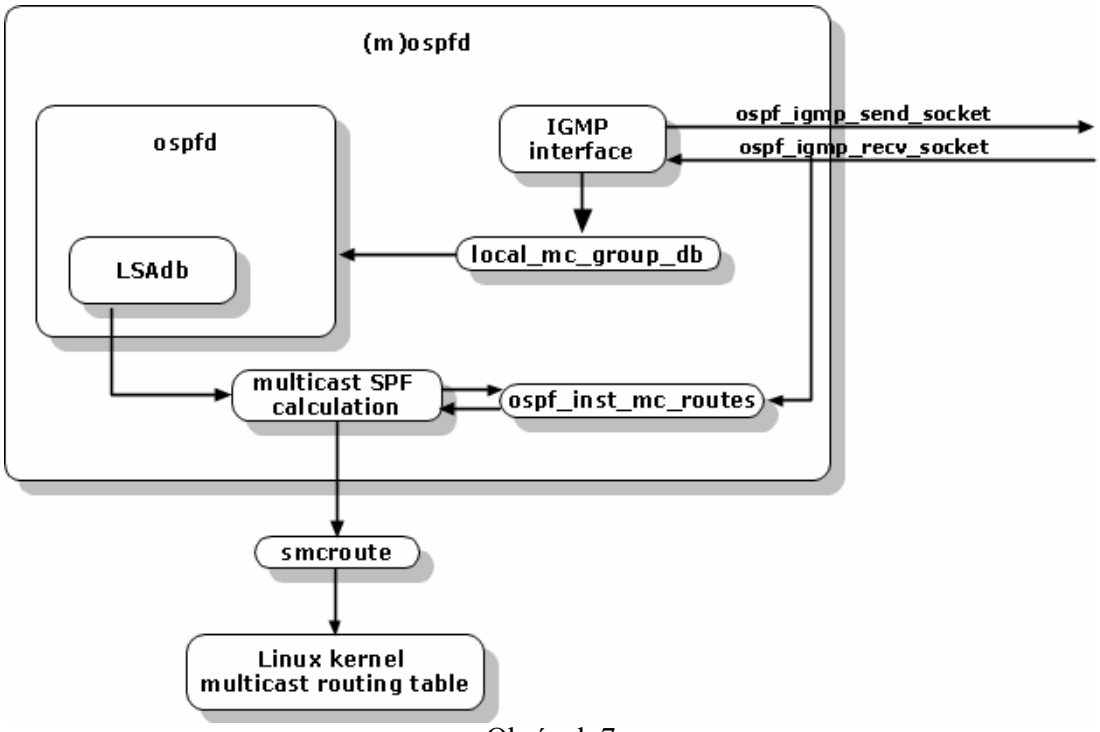
```
Multicast group     |Interface address
=====================================
234.111.111.111     | 192.168.1.1/24
234.111.111.222     | 192.168.2.1/24
```

4.2.5.2 Show ip ospf mc_cache_route

Vypíše obsah tabuľky *local_mc_group_db*, teda dvojice zdrojová adresa a cieľová multicastová adresa, pre ktoré existuje záznam v multicastovej routovacej tabuľke. Príklad výstupu:

```
Source address      |Dest mc address
=====================================
192.168.4.3         | 234.111.111.111
192.168.1.1         | 234.111.111.222
```

4.2.6 Nová architektúra



Obrázok 7

Na obrázku 7 je znázornená nová architektúra modulu ospf. Na obrázku vidieť, že sme využili existujúcu databázu LSA.

4.2.7 Obmedzenia, možné zlepšenia

Naša implementácia má niekoľko nedostatkov:

- nepodporuje pokročilé funkcie IGMPv3. Implementovaná funkcionalita je postačujúca na jednoduché pridanie stanice ku skupine, v ktorej je členom. Väčšina staníc nevyužíva žiadne ďalšie možnosti IGMP.
- pracuje len v rámci jednej AS a OSPF oblasti. Toto možno nie je chyba našej implementácie, ale skôr chyba vybranej technológie (riešenia), resp. naše riešenie nie je určené na efektívne multicastové routovanie medzi AS a OSPF oblasťami.

Ďalšie možné úpravy alebo zlepšenia by mohli byť:

- BPF filter, ktorý filtruje dáta z ospf_igmp_recv_socket-u rozšíriť tak, aby odfiltrovával aj multicastové dáta, pre ktoré existuje v kerneli cesta. Toto by vyžadovalo vytvoriť rozsiahlu logiku na úpravu BPF filtra tak, aby bola zachovaná efektivita
- Konfigurácia IGMP a multicastových časovačov (timeoutov) priamo z konzoly. Teraz sú dané v kóde ako premenné preprocesora.
- Zlepšenie obsluhy smcroute utility. Využiť všetky možnosti existujúceho API pre smcroute.

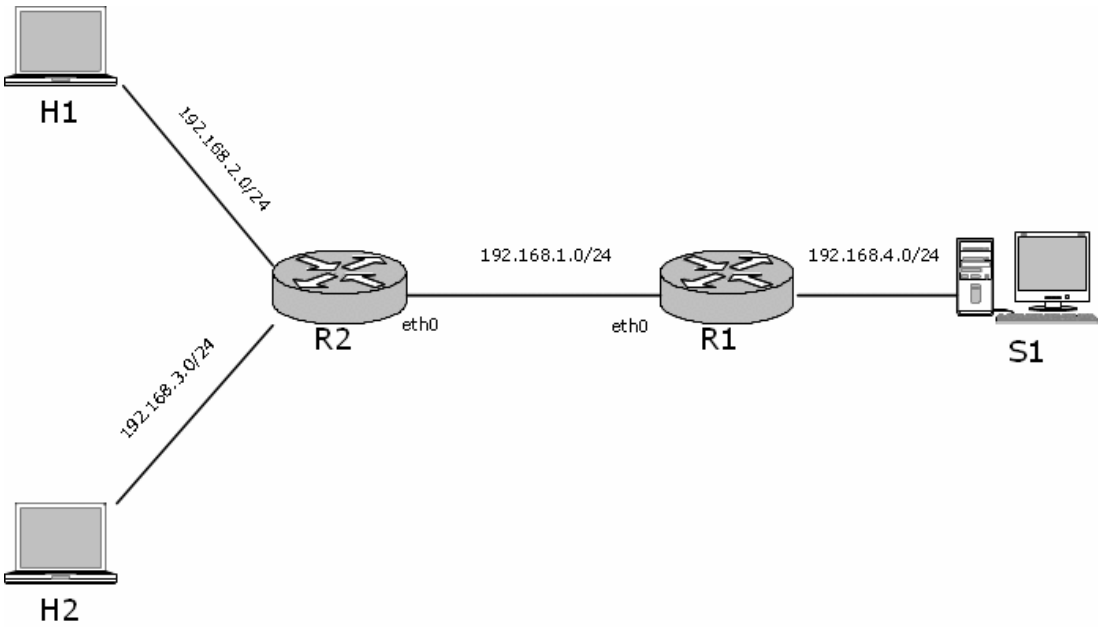
4.3 QoS

V linuxovom kerneli sú už QoS techniky implementované. Sú súčasťou balíka *iptables*. Ako konfiguračný nástroj sa používa *tc*. Konfigurácia cez tc prebieha cez príkazový riadok a pri náročnejších QoS nastaveniach je veľmi neprehľadná.

Preto bol vyvinutý nástroj *tcng (Traffic Control Next Generation)*, ktorý prináša vlastný jazyk na popis QoS nastavení tc. Tento jazyk je zrozumiteľný a prehľadný. Konfigurácie sa zapisujú do súboru. Tcng vie vygenerovať z takéhoto konfiguračného súboru dávku príkazov pre tc.

4.4 Testovanie

Pre testovanie sme sa snažili vytvoriť zapojenie, ktoré je možné nájsť aj v reálnom svete. Vyskúšali sme zapojenie dvoch routrov, pričom na jednom z nich bude pripojený server a na druhom dve stub siete, ktoré môžu potencionálne obsahovať viacero klientskych počítačov zapojených pomocou switchu. My sme v každej stub sieti mali jeden počítač. Zapojenie je znázornené na obrázku 8.



Obrázok 8

Testovali sme prenos videa súčasne s prenosom FTP. Cieľom bolo zabezpečiť, aby FTP prenos nazasahoval a nerušil prenos videa. To isté zapojenie sme testovali bez použitia aj s použitím QoS služieb.

Na prenos videa sme použili nástroj *VLC media player* (www.videolan.org). Ide o voľne šíriteľný program na prehrávanie a streamovanie videa. Umožňuje zastupovať klientskú aj serverovú časť. VLC media player používa protokol IGMPv3 na ohlasovanie prítomnosti multicasových skupín.

Video sme prenášali zo servera S1 na multicastovú adresu 234.111.111.111 a port 1234. Naše video malo priemernú prenosovú rýchlosť 278 kbs.

Na meranie prenosových rýchlostí sme použili *tshark.*.

QoS sme nastavili tak, že sme zapli na rozhraní eth0 routra R1 schedulovanie fronty pomocou tc. Použili sme scheduler HTB (http://luxik.cdi.cz/~devik/qos/htb/). Tento umožňuje vytvárať hierarchickú štruktúru tried dát. Pre každú triedu umožňuje stanoviť niekoľko parametrov. V našom teste sme použili dva:

- *rate* – prenosová rýchlosť, ktorá by mala byť vždy prístupná tejto triede
- *ceil* – maximálna prenosová rýchlosť, akou môžu byť dáta v danej triede prenášané

Ak existuje voľná prenosová kapacita na linke, táto je prerozdeľovaná medzi jednotlivé triedy v pomere, aký určujú ich rate parametre. Zaradenie paketov do tried je funkčnosť nástroja tc. Ten umožňuje využívať väčšinu políčiek TCP/IP protokolov.

V našom teste sme obmedzili celé rozhranie eth0 na 400kbs.

Trieda	Kritérium výberu paketov	HTB rate	HTB ceil
video	cieľový UDP port je 1234	350 kbs	400 kbs
ftp	zdrojový alebo cieľový TCP port je 20	30 kbs	400 kbs
other	Iné	20 kbs	400 kbs

Tabuľka 4

Týmito nastaveniami docielime, že v prípade, že je sieťou prenášané video (s prenosovou šírkou približne 300kbs), toto má prednosť pred ostatnými dátami. V prípade ak video prenášané nie je, je celá prensová šírka (400kbs) uvoľnená ostatným dátam. Naša konfigurácia môže byť v niektorých situáciách nevýhodná, pretože v tomto prípade majú ftp prenosy nárok na 3/5 prenosovej šírky.

Test sme spúšťali najprv so zapnutými QoS službami, potom s vypnutými, aby sme mohli sledovať výhody QoS. V prípade testu bez použitia služieb QoS sme obmedzili iba celkovú prenosovú rýchlosť pre všetky dáta na 400 kbs.

4.4.1 Konfigurácia, nastavenia, inštalácia

4.4.1.1 Systémové nastavenia

Sieťové rozhrania na routroch aj staniciach sme nastavili podľa obrázku 8. Na všetkých routroch bolo ďalej nutné:

- nainštalovať kernel, ktorý podporuje multicastové routovanie. Pri kompilácii zapnúť *CONFIG_IP_MULTICAST*

- toto routovanie zapnúť. V Debiane pridať do */etc/network/options* riadok *ip_forward=yes*

Ďalej sme museli nainštalovať utilitu *smcroute*.

Konfiguračné súbory sieťových rozhraní jednotlivých routrov určené pre operačný systém Debian sú v prílohe A.1.

4.4.1.2 Quagga

Všetky siete sme pridali do OSPF domény príkazmi *network*. Rozhrania, ktoré sú pripojené na stub siete sme nastavili ako pasívne (*passive-interface*). To zabezpečí, že nebudú zbytočne posielať OSPF Hello pakety do týchto sietí. Konfiguračné súbory zebra a ospfd modulov sa nachádzajú v prílohe A.2.

4.4.1.3 Tcng

QoS sme nastavili na rozhraní eth0 routra R1 podľa tabuľky 4. Konfiguračné súbory pre tcng sú v prílohe A.3.

4.4.2 Priebeh testu

Pri obidvoch testoch (aj s použitím QoS aj bez) sme postupovali v nasledujúcich krokoch:

1. pustili sme video prenos cez VLC, teda spustili sme video klientov aj video server
2. pustili sme *tshark* na eth0 rozhraní routra R2
3. po približne 20 sekundách sme spustili ftp prenos z jedného klientského počítača na server
4. po ďalších približne 90 sekundách sme prenos videa vypli

Celkovo meranie prebiehalo 120 sekúnd.

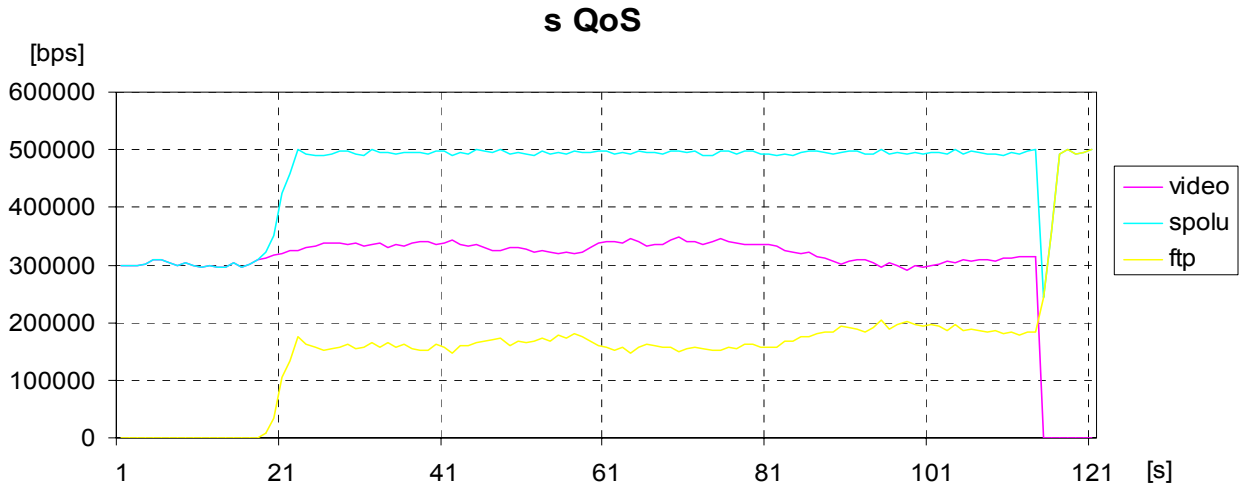
4.4.3 Meranie

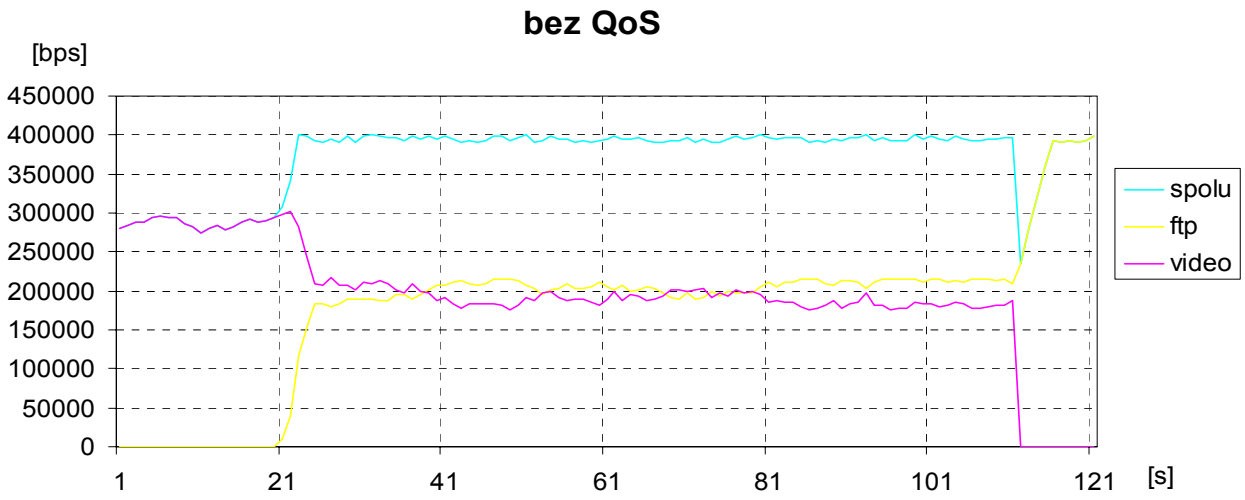
Výsledok merania testu pre nás je graf rozloženia prenosovej kapacity siete, ktorá spája routre R1 a R2. Merali sme prenosové šírky tried definovaných pomocou tcng nástroja, teda tried, ktoré mali definované QoS parametre prenosu. Keďže do triedy *other* spadali iba nenáročné SSH prenosy nutné na konfiguráciu a monitorovanie routrov takisto nenáročná signálna časť FTP prenosu, túto triedu sme nemerali.

Merali sme pomocou utility *tshark*, čo je konzolová verzia programu *wireshark* (pôvodne nazývaný *Ethereal*). Výstup sme previedli do formy zoznamu, kde je uvedená prenosová rýchlosť pre danú triedu v jednotkách bps. Podrobná tabuľka s hodnotami sa nachádza v prílohe A.4.

4.4.4 Grafy

Výstupné dáta sme zhrnuli do grafov. Prvý graf ukazuje analýzu dát so zapnutým QoS, druhý bez. Vyjadrujú závislosť prenosovej rýchlosti jednotlivých tried od času.





4.4.5 Výsledky

4.4.5.1 QoS

Z grafov vidíme, že pri použití služieb QoS sa prenosová rýchlosť nejako výrazne nezníži po začatí FTP prenosu. Pri teste bez použitia QoS sme zaznamenali viditeľnú zmenu kvality prenášaného videa po spustení FTP prenosu. To sa potvrdilo aj štatistikami zahodených framov videa.

Ďalej je vidieť, že v situácii s QoS využíva FTP spojenie nielen priradených 20kbs, ale celú voľnú prenosovú šírku, ktorú nevyužíva video prenos.

4.4.5.2 Multicast

Vidíme, že aj keď je video prijímané obidvoma stanicami H1 a H2, je prenášané medzi routrami R1 a R2 iba jedenkrát.

4.4.5.3 Reálne riešenie

Situácia obdobná ako pri našom zapojení môže nastať v prípade spoločnosti, ktorá má dve pobočky a chce medzi nimi môcť prenášať aj videokonferencie. Jednu pobočku tvorí stub sieť so serverom S1 a druhú tvoria stub siete so stanicami H1 a H2. Tieto môžu predstavovať dve oddelenia.

Linka medzi routrami R1 a R2 môže byť typu WLAN, práve potom vzniká problém s nedostatočnou kapacitou na prenos videa zároveň s inými dátami. Takisto sa môže jednať o prenajatú linku od verejného ISP. V tomto prípade by bolo treba zabezpečiť si aj v rámci siete ISP prioritizáciu dát. Ak využíva ISP služby DiffServ, môžeme pridať označovanie paketov na základe našich tried políčkom DSCP. Potrebujeme takisto s ISP uzavrieť zmluvy SLA o zabezpečení QoS parametrov vybraným službám.

5 Záver

V dnešnej dobe existuje viacero protokolov a architektúr podporujúcich QoS a multicasty. Niektoré využívajú a podporujú tieto technológie v širšom zmysle (IntServ, MOSPF), iné len povrchne (DiffServ, DVMRP). Prvá skupina sa potýka s problémami škálovateľnosti a ťažkosťami pri implementácii a nasadení, druhá s nepresnosťou údajov, nutnosťou abstrakcie a tým pádom stratou kvality samotnej funkčnosti.

Postupom času sa ukazuje, že práve technológie, ktoré podporujú QoS a multicastové technológie len v obmedzenej podobe. Takáto funkčnosť sa ukazuje ako dostatočná, čo spolu s nízkymi časovými a finančnými nákladmi na implementáciu a nasadenie prispelo aspoň k čiastočnému masovému rozšíreniu týchto technológií.

Väčšina sieťových zariadení, či už sú to linuxové stroje alebo špičkové zariadenia od výrobcov sieťových technológií, má implementované QoS funkcie aspoň v podobe usporiadania paketov do front na základe ich cieľovej a zdrojovej adresy prípadne portu.

Podobne multicastové routovanie je implementované v sieťových zariadeniach. Situácia s linuxovými strojmi je prakticky obmedzená na DVMRP. Tento protokol ale stačí na to, aby výrazne znížil zbytočné zaťaženie siete.

Ďalší vývoj v týchto oblastiach bude smerovať napr. ku skúmaniu QoS mechanizmov pre bezdrôtové ad-hoc siete, hľadaní optimálnych algoritmov pre rôzne QoS obmedzenia multicastových prenosov atď.

S príchodom IPv6 sa situácia zmení. Pri návrhu protokolu IPv6 bolo myslené na QoS a multicasty. Po jeho rozšírení sa možno konečne dočkáme masového rozšírenia pravého QoS a multicastu.

6 Použitá literatúra

[ALGS] – J. Jaffe, „Algorithms for finding paths with multiple constraints“. Networks, vol. 14.

[IBM QOS] – Fellows, J., Straeten, D., “Aplication-Driven Netoworking: Class of Service in IP, Ethernet, and ATM networks”, IBM, December 1999.

[IGMP] – Fenner, W., Xerox PARC, “Internet Group Management Protocol, Version 2”, RFC 2236, November 1997.

[MC DS] – Wang, Y., Kantola, R., Liu, S., Adding Multi-class Routing into DiffServ Architecture”, ICW’05.

[MC HOWTO] – Goyeneche, “Multicast over TCP/IP HOWTO”.

[MC INV] – Manolov, N., Gamil, A., Wong, S., “An investigation into Multicasting”.

[MC TU] – Laxman, H., Mukherjee, B., Mukherjee, S., “Multicast Routing Algorithms and Protocols: A Tutorial”, IEEE network 2000.

[MPLS DIFFSERV] – Singh, A., “QoS and Traffic Engineering: MPLS, DiffServ and Constraint Based Routing”, May 2000.

[MPLS TE DIFFSERV] – Minei, I., „MPLS DiffServ-aware Traffic Engineering“, Juniper Networks, 2004.

[QOS FR] – Crawley, E., Nair, R., Rajagopalan, B., Sandick, H., “A Framework for QoS-based Routing in the Internet”, RFC 2386, August 1998.

[QOS OV] – Deng, B., Liu, H., Zheng, M., “Overview of QoS Routing”.

[QOS TU] – Chimento, P.F., “Tutorial on QoS support for IP”.

[RFC IP] – Postel J., “Internet Protocol”, RFC 791, USC/Information Sciences Institute, September 1981.

[RFC LDP] – Andersson, L., "LDP Specification", RFC 3036, Nortel Networks Inc., January 2001.

[RFC OSPF] – Moy, J., "OSPF Version 2", RFC 1583, Proteon, Inc., March 1994.

[RFC MOSPF] – Moy, J., "Multicast Extensions to OSPF", RFC 1584, Proteon, Inc., March 1994.

[RFC QOSPF] – G. Apostolopoulos, D. Williams, S. Kamat, R. Guerin, A. Orda, T. Przygienda, “QoS Routing Mechanisms and OSPF Extensions”, RFC 2676, August 1999.

[StupidNet] – David Isenberg, Rise of the Stupid

7 Prílohy

A.1 Konfiguračné súbory sieťových rozhraní

V operačnom systéme Debian sú tieto súbory uložené ako */etc/network/interfaces*

A.1.1 Konfiguračné súbory pre router R1

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
    address 192.168.1.2
    netmask 255.255.255.0

auto eth1
iface eth1 inet static
    address 192.168.4.1
    netmask 255.255.255.0
```

A.1.2 Konfiguračné súbory pre router R2

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
    address 192.168.1.1
    netmask 255.255.255.0

auto eth1
iface eth1 inet static
    address 192.168.2.1
    netmask 255.255.255.0

auto eth2
iface eth2 inet static
    address 192.168.3.1
    netmask 255.255.255.0
```

A.2 Konfiguračné súbory Quaggy

Umiestnenie konfiguračných súborov Quaggy (zebra.conf, ospfd.conf) je závislé od parametra *sysconfdir* použitého pri kompilácii, resp. spúšťaní skriptu *configure*.

A.2.1 Konfiguračné súbory modulu zebra

A.2.1.1 Router R1

```
log stdout
password heslo
!
interface eth0
description ISP1
multicast
!
interface eth1
description Branch1
multicast
```

A.2.1.2 Router R2

```
log stdout
password heslo
!
interface eth0
description ISP1
multicast
!
interface eth1
description Branch2_Dep1
multicast
!
interface eth2
description Branch2_Dep2
multicast
```

A.2.2 Konfiguračné súbory modulu ospfd

A.2.2.1 Router R1

```
!
password heslo
log stdout
!
interface eth0
!
interface eth1
!
router ospf
 network 192.168.1.0/24 area 0.0.0.0
 network 192.168.4.0/24 area 0.0.0.0
 passive interface eth1
```

```
!
line vty
!
```

A.2.2.2 Router R2

```
!
password heslo
log stdout
!
interface eth0
!
interface eth1
!
interface eth2
!
router ospf
 network 192.168.1.0/24 area 0.0.0.0
 network 192.168.2.0/24 area 0.0.0.0
 network 192.168.3.0/24 area 0.0.0.0
 passive interface eth1
 passive interface eth2
!
line vty
!
```

A.3 Konfiguračný súbor tcng

A.3.1 Test s QoS

```
#include "fields.tc"

#define INTERFACE  eth0

dev INTERFACE {
    egress {

        class ( <$video> )    if udp_dport == 1234 ;
        class ( <$ftp> )  if tcp_sport == 20 || tcp_dport == 20 ;
        class ( <$other> )  if 1 ;

        htb () {
            class ( rate 400kbps, ceil 400kbps ) {
                $video   = class ( rate  350kbps, ceil 400kbps ) {
sfq; } ;
                $ftp  = class ( rate 30kbps, ceil 400kbps ) { sfq; } ;
                $other = class ( rate 20kbps, ceil 400kbps ) { sfq; }
;
            }
```

```
        }
    }
}
```

A.3.2 Test bez QoS

```
#include "fields.tc"

#define INTERFACE  eth0

dev INTERFACE {
    egress {

        class ( <$all> )  if 1 ;

        htb () {
            class ( rate 400kbps, ceil 400kbps ) {
                $all   = class ( rate  400kbps, ceil 400kbps ) { sfq;
} ;
            }
        }
    }
}
```

A.4 Výsledky merania

A.4.1 Test s QoS

sec	video	ftp	sum
0	298552	0	298552
1	297864	0	297864
2	299016	0	299016
3	302088	0	302088
4	308976	0	308976
5	309792	0	309792
6	303416	0	303416
7	299928	0	299928
8	304696	0	304696
9	299536	0	299536
10	296144	0	296144
11	298544	0	298544
12	296992	0	296992
13	296360	0	296360
14	303288	0	303288
15	297304	0	297304
16	301128	0	301128
17	308432	0	308432

18	312736	8994	321730
19	317856	34435	352291
20	320112	104960	425072
21	324488	134035	458523
22	325216	174260	499476
23	330832	161229	492061
24	333464	157327	490791
25	338688	152160	490848
26	338840	154841	493681
27	339272	157806	497078
28	334848	162856	497704
29	338904	154616	493520
30	333800	157027	490827
31	334872	164429	499301
32	337240	156850	494090
33	330360	164601	494961
34	334824	157154	491978
35	333800	161923	495723
36	339024	154934	493958
37	341504	152407	493911
38	340728	150660	491388
39	335336	163103	498439
40	338960	158329	497289
41	342072	147997	490069
42	334896	160955	495851
43	333632	158794	492426
44	334944	164842	499786
45	328968	168247	497215
46	325712	170064	495776
47	326024	173746	499770
48	330920	161132	492052
49	329224	166868	496092
50	326752	164876	491628
51	322128	168765	490893
52	323640	174017	497657
53	323128	168799	491927
54	318672	177691	496363
55	321776	171638	493414
56	318568	180294	498862
57	321352	174371	495723
58	328968	166607	495575
59	336944	159889	496833
60	341440	156335	497775
61	339704	151797	491501
62	337992	157037	495029
63	345944	147347	493291
64	340824	158205	499029
65	333520	161540	495060
66	336096	159274	495370
67	334736	156623	491359
68	342640	155996	498636
69	347984	150144	498128
70	341624	153386	495010
71	339624	157250	496874
72	335056	155377	490433
73	339424	151730	491154
74	347056	150985	498041

75	341008	156040	497048
76	337320	154678	491998
77	335280	161292	496572
78	335288	162884	498172
79	334104	157356	491460
80	334400	158488	492888
81	333536	156569	490105
82	325776	167992	493768
83	322000	168497	490497
84	319608	174296	493904
85	321184	176011	497195
86	315520	181113	496633
87	310816	183724	494540
88	306872	184504	491376
89	302152	192713	494865
90	307592	190107	497699
91	308688	189773	498461
92	308416	183808	492224
93	302784	189982	492766
94	295904	203819	499723
95	303048	189559	492607
96	297728	196230	493958
97	291528	201493	493021
98	298512	196612	495124
99	297088	194201	491289
100	298488	196401	494889
101	300560	195065	495625
102	306416	185112	491528
103	304024	195521	499545
104	308064	185264	493328
105	307232	189597	496829
106	308608	185699	494307
107	309200	182835	492035
108	307016	186854	493870
109	310488	180707	491195
110	311528	182639	494167
111	313656	179199	492855
112	314216	183128	497344
113	315040	184327	499367
114	0	244678	244678
115	0	358908	358908
116	0	493104	493104
117	0	499581	499581
118	0	491781	491781
119	0	495338	495338
120	0	499220	499220

A.4.2 Test bez QoS

sec	video	ftp	sum
0	279952	0	279952
1	283224	0	283224
2	288280	0	288280
3	288656	0	288656

4	293640	0	293640
5	296264	0	296264
6	293744	0	293744
7	293144	0	293144
8	285528	0	285528
9	281480	0	281480
10	273912	0	273912
11	281152	0	281152
12	284384	0	284384
13	278808	0	278808
14	282328	0	282328
15	287656	0	287656
16	292528	0	292528
17	287944	0	287944
18	290944	0	290944
19	293128	0	293128
20	297232	9698	306930
21	302008	39086	341094
22	282530	117341	399871
23	244865	152960	397825
24	209599	183840	393439
25	207824	183712	391536
26	216406	179040	395446
27	206618	183608	390226
28	207754	190216	397970
29	201167	190040	391207
30	210254	189368	399622
31	210074	189720	399794
32	212391	186744	399135
33	209108	187784	396892
34	201971	194984	396955
35	198090	195008	393098
36	209481	188504	397985
37	198608	195144	393752
38	197004	201352	398356
39	187604	206424	394028
40	191035	207456	398491
41	182831	212048	394879
42	178318	212464	390782
43	182714	209384	392098
44	183307	207792	391099
45	183748	208848	392596
46	183198	216000	399198
47	182248	216000	398248
48	175983	216000	391983
49	182418	214032	396450
50	192115	207584	399699
51	187701	202520	390221
52	197874	194544	392418
53	198404	201096	399500
54	191783	203616	395399
55	187301	208272	395573
56	188540	202352	390892
57	189719	203088	392807
58	184934	205496	390430
59	182409	211328	393737
60	189028	205344	394372

61	198638	200408	399046
62	187002	207160	394162
63	196086	199280	395366
64	193953	202208	396161
65	186562	205888	392450
66	188979	202552	391531
67	193991	197136	391127
68	200875	191936	392811
69	201778	190328	392106
70	199018	196880	395898
71	201083	189416	390499
72	202457	192168	394625
73	192297	198792	391089
74	196711	194288	390999
75	194037	199840	393877
76	200577	198272	398849
77	197489	196632	394121
78	199383	198272	397655
79	194416	205320	399736
80	185053	211360	396413
81	187912	205888	393800
82	186057	211032	397089
83	184804	211600	396404
84	180030	216000	396030
85	175393	215808	391201
86	177086	216000	393086
87	181915	208744	390659
88	187258	206568	393826
89	177953	213824	391777
90	183010	213744	396754
91	185227	210664	395891
92	196383	203336	399719
93	180648	211328	391976
94	181284	215528	396812
95	176172	216000	392172
96	177081	216000	393081
97	177536	216000	393536
98	185521	214384	399905
99	183856	210264	394120
100	183127	215280	398407
101	178656	216000	394656
102	182020	210584	392604
103	184704	213936	398640
104	182951	211656	394607
105	177099	216000	393099
106	177518	216000	393518
107	179068	216000	395068
108	180707	213352	394059
109	180760	216000	396760
110	187180	208600	395780
111	0	234725	234725
112	0	279643	279643
113	0	319909	319909
114	0	359972	359972
115	0	392752	392752
116	0	391517	391517
117	0	391832	391832

```
118     0       390869  390869
119     0       392194  392194
120     0       398736  398736
```