

Tímový projekt
Prípadová štúdia pre VoIP sieť
chiNETeam (11)

Tím č.: 11
Akademický rok: 2009/2010

Filip Burda, Bc.
Peter Havrila, Bc.
Marián Knězek, Bc.
Klaudia Konôpková, Bc.
Juraj Nemeček, Bc.
Ján Murányi, Bc.

Obsah

Zadanie.....	1
1 Úvod.....	2
1.1 Motivácia.....	2
1.2 Prehľad dokumentu.....	3
2 Protokol SIP a protokol SDP.....	4
2.1 Úlohy protokolu SIP.....	4
2.2 Protokol SIP a model RM OSI.....	4
2.3 Základný hovor.....	5
2.3.1 Identifikácia používateľa.....	5
2.4 Všeobecné hlavičky.....	6
2.4.1 Pole Call-ID.....	6
2.4.2 Pole Cseq.....	6
2.4.3 Pole From.....	6
2.4.4 Pole To.....	6
2.4.5 Pole Via.....	7
2.4.6 Pole Content-Type.....	7
2.4.7 Pole Content-Length.....	7
2.5 Odpovede protokolu SIP.....	8
2.6 SIP a nespôľahlivé protokoly.....	8
2.6.1 Nie INVITE transakcie.....	8
2.6.2 INVITE transakcie.....	9
2.7 Protokol SDP.....	9
3 IP Multimedia Subsystem.....	12
3.1 Architektúra IMS.....	13
3.1.1 IMS entity a funkcie.....	13
3.1.2 Call Session Control Function (CSCF).....	14
3.1.3 Databázy.....	16
3.1.4 Služby.....	17
3.1.5 Vnútorne funkcie.....	17
3.1.6 Podporné funkcie.....	18
4 Optimalizácia siete.....	19
4.1 Topológie.....	19
4.2 Zariadenia.....	20
5 Smerovanie v TCP/IP.....	21
5.1 Klasifikácia smerovania.....	21
5.1.1 Statické smerovanie.....	21
5.1.2 Dynamické smerovanie.....	21
5.2 Klasifikácia smerovacích algoritmov a protokolov.....	22
5.2.1 Aspekt rozsahu použitia.....	22
5.2.2 Aspekt smerovacieho algoritmu.....	23
5.2.3 Aspekt transportného mechanizmu.....	23
5.3 Protokoly s DV algoritmom.....	24
5.3.1 Protokol BGP.....	24
5.4 Protokoly s LS algoritmom.....	25

5.4.1	Protokol OSPF.....	27
5.4.2	Protokol IS-IS.....	30
6	Konvergencia na sieťovej vrstve.....	33
6.1	Konvergencia v sieti.....	34
6.2	Časové parametre smerovacích protokolov.....	35
6.2.1	Protokol OSPF.....	36
6.2.2	Protokol IS-IS.....	37
6.2.3	Protokol RSVP.....	38
6.2.4	Protokol LDP.....	38
6.3	Stabilita po časovej optimalizácii.....	39
7	Kvalita služieb.....	40
7.1	Potreba kvality služieb.....	40
7.2	Nedostatok transportnej kapacity.....	40
7.3	Oneskorenie.....	41
7.3.1	Variabilné oneskorenia.....	42
7.4	Straty v toku dát.....	42
7.5	Modely prístupu ku kvalite služieb.....	42
7.5.1	Model Best Effort.....	42
7.5.2	Model IntServ.....	43
7.5.3	Model DiffServ.....	43
7.6	Nástroje Differentiated Services.....	44
7.6.1	Značenie paketov.....	44
7.6.2	Manažment radov.....	45
7.6.3	Techniky predchádzania zahltenia.....	45
8	Redundancia na sieťovej vrstve.....	47
8.1	Protokol VRRP.....	47
8.1.1	Terminológia v protokole VRRP.....	47
8.1.2	Príklad použitia.....	48
9	Technológia MPLS.....	49
9.1	Úvod do MPLS.....	49
9.2	MPLS návestia.....	50
9.3	Vnorené MPLS návestia.....	50
9.4	MPLS smerovanie.....	51
10	MPLS Traffic Engineering.....	53
10.1	Využitie MPLS TE.....	54
10.1.1	Atribútové bity.....	54
10.1.2	Administratívna váha.....	54
10.1.3	Manažment šírenia stavových informácií.....	55
10.1.4	Opätovná optimalizácia tunelov.....	55
10.1.5	Rozkladanie záťaže.....	55
10.1.6	Posunutie susedstva.....	56
10.1.7	Automatické určenie šírky pásma.....	56
10.1.8	Ochrana.....	56
10.1.9	Automatická cesta.....	57
10.2	Kvalita služieb v MPLS TE.....	57
10.3	MPLS TE s MPLS VPN a MPLS QoS.....	58
11	Protokol IPv6.....	59

12 Zhodnotenie analýzy.....	61
13 Špecifikácia požiadaviek.....	62
14 Návrh.....	63
14.1 Platforma ESDS na vývoj aplikácii v IMS.....	64
15 Prototyp riešenia.....	67
15.1 Topológia v IMS podsieti	67
15.1.1 CSCF.....	68
15.1.2 HSS.....	69
15.1.3 Internetová brána a DNS server.....	70
15.1.4 Asterisk.....	71
15.2 Transportná sieť.....	72
15.2.1 Protokol VRRP.....	75
15.2.2 Protokol IS-IS.....	75
15.2.3 MPLS.....	75
15.3 Zhodnotenie.....	76
Slovník skratiek.....	77
Literatúra.....	79

Zadanie

IP Multimedia Subsystem je v súčasnosti najperspektívnejšia technológia pre implementáciu v prostredí telekomunikačných operátorov.

Cieľom projektu je vytvoriť prípadovú štúdiu zameranú na konkrétny aspekt tejto siete (zabezpečenie kvality služby, bezpečnosť a pod.)

V rámci tímového projektu bude nutné vybrať cieľovú platformu (v prípade IMS napr. OpenIMS core), navrhnúť zapojenie celej siete, pripraviť scenáre a metriky a nakoniec dosiahnuté výsledky vyhodnotiť.

Prirodzenou súčasťou práce je administrácia platformy a vytvorenie nielen dokumentácie k projektu, ale aj celej prípadovej štúdie vo vhodnom formáte.

Vedúci tímu Ing. I. Kotuliak, PhD.

1 Úvod

Milý čitateľ,

dostáva sa Vám do rúk dokument, ktorý vznikol spoločnou prácou skupiny mladých študentov na predmete Tímový Projekt na Fakulte Informatiky a Informačných Technológií Slovenskej Technickej Univerzity. Témou tejto práce je prípadová štúdia pre VoIP sieť, ktorú naša skupina analyzovala, navrhla a vytvorila z pohľadu zameranom na zabezpečovanie kvality služieb pri udalostiach s nepriaznivým vplyvom na sieť. V aktuálnom stave tohto dokumentu sa nachádza analýza celkovej problematiky z oblasti počítačových sietí a VoIP spracovaná jednotlivými členmi tímu. V analýze sa nachádza popis postupov, technológií a hlavne potrieb, ktoré vyžadujú moderné multimediálne relácie vystavané na báze IP Multimedia Subsystem a ďalej spracovanie možností zabezpečiť požiadavky na transportnú sieť s maximálnym dôrazom na robustnosť siete pri zabezpečovaní riadenia kvality služieb. Tu sa tím zameril na aktuálne najprogressívnejšiu technológiu používanú poskytovateľmi služieb a to *Multi-Protocol Label Switching* (MPLS) a možnosti riadenia kvality služieb pomocou pridruženej technológie *MPLS Traffic Engineering* (MPLS TE).

Veríme, že tento dokument splní očakávania čitateľa a tiež posluží ako cenné zhrnutie uvedených technológií.

Prijemné čítanie praje
chiNETeam

1.1 Motivácia

Ako napovedá zadanie projektu, IP Multimedia Subsystem je v dnešnej dobe najatraktívnejším systémom na prevádzku VoIP a multimediálnych služieb pre komerčných poskytovateľov služieb. Vychádzajúc z profesionálneho zázemia členov tímu, ktoré pokrýva VoIP implementácie, ale aj problematiku dátových transportných sietí, bolo veľmi apelujúce uskutočniť prípadovú štúdiu, v ktorej spracujeme práve spoluprácu týchto dvoch neoddeliteľných vrstiev pri implementácii celkovej infraštruktúry služieb. Pre všetkých členov tímu umožní tento projekt, okrem adekvátneho profesionálneho rastu, zároveň aj pohľad na budovanie komplexných IP sietí a VoIP služieb.

Tému si vybral celý tím jednohlasne, nakoľko sa všetci pohybujeme vo svete počítačových sietí. Jednotliví členovia majú pracovné skúsenosti z oblasti počítačových sietí, ich nasadzovania, správy a dokonca aj výučby.

Tím má záujem rozširovať si svoje vedomosti v oblasti konvergovaných sietí, NGN a IMS, nakoľko je toto oblasť pre nás atraktívna. Vytvorenie prípadovej štúdie je pre nás vynikajúcou príležitosťou ako naplniť naše osobné ciele a to rozšíriť si obzory v tejto oblasti a zároveň vytvoriť zaujímavý produkt. Je to dobrá príležitosť ako sa zamerať aj na iné technológie a nielen technológie a produkty od firmy Cisco, s ktorými sme sa doteraz stretávali. Zároveň máme k dispozícii ten najväčší kapitál a to sú naše vedomosti a chuť učiť sa nové. Myslíme si, že pre danú tému máme výborné predpoklady, nakoľko všetci sme aktívnou súčasťou Cisco akademie a vieme využiť naše

bakalárske práce, vedecké publikácie a školské projekty na vytvorenie výsledného produktu. Naše doterajšie práce pri spojení vyvolávajú silné synergické efekty.

1.2 Prehľad dokumentu

Táto práca je analýzou k prípadovej štúdii VoIP siete. Dokument pozostáva zo 14 kapitol. Prvá kapitola je úvodom k tomuto dokumentu. V druhej kapitole si predstavíme protokol SIP a protokol SDP. V kapitole 3 rozoberáme IP Multimedia Subsystem, jeho architektúru a súčasti. Úvod do optimalizácie siete sa nachádza v kapitole 4. Piata kapitola samostatne popisuje smerovanie v TCP/IP sieťach. V kapitole 6 rozoberáme konvergenciu v sieti a časové parametre smerovacích protokolov. Kapitola 7 sa venuje problematike kvality služieb – QoS. Ôsma kapitola sa venuje redundancii na sieťovej vrstve. Kapitola 9 opisuje technológiu MPLS a nasledujúca kapitola sa venuje jednej jej súčasti – MPLS TE. V kapitole 11 diskutujeme o možnostiach protokolu IPv6 v medziach nášho projektu. Kapitola 12 je zhodnotením analýzy. Požiadavky špecifikujeme v kapitole 13. Návrhu riešenia sme umiestnili do kapitoly 14. Prototyp riešenia s počiatočnou implementáciou je v nasledujúcej kapitole 15. Práca sa končí zoznamom skratiek použitých v dokumente a použitou literatúrou.

Táto práca vyžaduje od čitateľa aspoň základné znalosti z oblasti počítačových sietí. Práca vychádza zo spojenia viacerých našich bakalárskych prác a vedeckých článkov, odkiaľ sú prevzaté celé state. Práca sa orientuje najmä na zbernicovú topológiu, konkrétne *ethernet*, keďže je to jedna z najpoužívanejších prístupových metód. Práca sa nebude bližšie venovať bezdrôtovým riešeniam. Jadro problematiky rozoberáme najmä na sieťovej vrstve a na úrovni aplikácií.

V texte sa bude nachádzať niekoľko anglických pojmov, ktoré nebudeme prekladať do ich slovenského ekvivalentu z dôvodu nejednoznačnosti a sťaženej prehľadnosti. Pôjde najmä o pojmy z oblasti IMS.

Anglické pojmy sú označené *kurzívou*. Podstatné informácie sú vyznačené **tučným** písmom. K preloženým pojmom uvádzame aj ich anglický ekvivalent. Všetky skratky obsiahnuté v texte je možné vyhľadať v slovníku pojmov na strane 77.

2 Protokoly SIP a protokol SDP

Protokol SIP je v dnešnej dobe najpoužívanejším protokolom v oblasti IP telefónie. Prvýkrát bol predložený Markom Handley a Evou Scholler ako IETF draft vo februári 1996 s názvom *Session Invitation Protocol*. Súčasne s protokolom SIPv1 bol uvedený protokol SCIP (*Simple Conference Invitation Protocol*) taktiež ako draft. Neskôr tieto dva protokoly vytvorili SIPv2. Zmenený bol názov na *Session Initiation Protocol*. Protokol bol postavený na protokole HTTP (*Hyper Text Transport Protocol*). Až vo februári 1999 dosiahol požadovanú úroveň a mohol byť uverejnený ako [RFC2543]. V júni 2002 pod vedením Jonathana Rosenberga bol označený ako prekonaný a bol nahradený [RFC3261, RFC2396].

2.1 Úlohy protokolu SIP

Protokol SIP je signalizačným protokolom. Neumožňuje prenos multimediálnych dát. Pri vývoji tohto protokolu sa dbalo, aby výsledný štandard bol veľmi jednoduchý a ľahko implementovateľný. Z tohto dôvodu v ňom boli zadefinované základné požiadavky, ktoré má umožňovať tento protokol:

- **Určenie polohy** – zistenie základných technických informácií ako IP adresa pre zabezpečenie komunikácie, ako medzi používateľskými stanicami, tak aj medzi serverom a klientom
- **Zabezpečenie dostupnosti** – zaručenie, že koncové zariadenie bude dostupné a schopné komunikovať
- **Vlastnosti koncových zariadení** – zistenie možných vlastností koncového zariadenia na prenos informácií
- **Zriadenie relácie** – dohodnutie parametrov oboch volajúcich strán, signalizácia a nadviazanie hovoru
- **Spravovanie relácie** – prenos, ukončenie a udržiavanie relácie (zmena prípadných vlastností prenosu) [1]

2.2 Protokol SIP a model RM OSI

Protokol SIP je definovaný aj nad transportným protokolom TCP aj nad protokolom UDP. V praxi sa častejšie používa protokol UDP, hlavne pre jeho vlastnosti, ktoré umožňujú malé oneskorenie. Napriek tomu RFC 3261 odporúča použitie protokolu TCP ako spoľahlivého protokolu. Predchádzajúce RFC 2543 odporúčalo protokol UDP. Táto zmena bola prijatá hlavne z dôvodu prenosu väčších správ. Hlavná nevýhoda UDP prenosu je problematická fragmentácia správ pri ich väčšom rozsahu. Preto, aj keď stanice komunikujú hlavne prostredníctvom UDP protokolu, je dôležité, aby boli schopné využívať aj protokol TCP. V prípade použitia protokolu UDP je možné v jednom datagrame poslať iba jednu SIP správu. Pri použití TCP spojenia rozdelenie správ určuje *Content-Length* metóda v SIP správe.

Štandardný TCP alebo UDP port, ak sa priamo nežiada inak, na ktorom aplikácie počúvajú prichádzajúce SIP správy, je 5060. V prípade použitia enkapsulácie TLS (*Transport Layer Security*) protokolu je to port 5061.

2.3 Základný hovor

Hovor, alebo dialóg, sa skladá z transakcií a tie sa skladajú z požiadaviek. Klienti teda komunikujú prostredníctvom transakcií. Každá transakcia je začatá počiatočnou požiadavkou a ukončená finálnou odpoveďou. V prípade štandardného hovoru, keď Anna (volajúca) volá Borisa (volaný), vyšle na známu adresu požiadavku *INVITE*. Boris môže odpovedať niekoľkými nepovinnými odpoveďami (*TRYING*, *RINGING*) a následne potvrdí prijatie hovoru finálnou odpoveďou *OK*. Touto odpoveďou sa končí prvá transakcia. Nasleduje druhá transakcia, keď Anna potvrdí prijatie hovoru. V tomto momente sa nadviaže relácia protokolu RTP (*Real-Time Protocol*), ktorý prenáša multimediálne informácie. Táto časť nie je transakciou, bola ňou iba dohodnutá. Ak sa Anna alebo Boris rozhodnú hovor ukončiť, pošlú druhej strane požiadavku *BYE* a tá ukončenie potvrdí finálnou odpoveďou *OK*.

Dialóg je definovaný rovnakými hodnotami polí *From*, *To*, *Call-ID* a rovnakou kombináciou polí *Via*. Po nadviazaní hovoru všetky správy musia obsahovať dohodnuté hodnoty týchto polí. Hodnota *CSeq* identifikuje transakciu. Jej hodnota je rovnaká pre všetky požiadavky a odpovede v rámci danej transakcie. Jej číslo sa nemení iba v prípade, že ide o transakciu *ACK*, ktorá prišla ako odpoveď na 2xx odpoveď, alebo je to transakcia *CANCEL*. V oboch prípadoch sa ale mení menová hodnota pol'a. Číslo zostáva rovnaké, aby bolo možné jednoducho vyhľadať, na ktorú transakciu prišlo dané *ACK* alebo *CANCEL*.

2.3.1 Identifikácia používateľa

Vo svete protokolu SIP sa jednotlivé entity identifikujú prostredníctvom SIP adresy známej ako URL (*Uniform Resource Locator*) alebo URI (*Uniform Resource Identifier*).

sip:alice@192.168.1.2

Táto adresa vychádza z RFC 2396 (*Uniform Resource Identifiers (URI): Generic Syntax*). Prvá časť určuje prekladaču URI, ako ju má správne interpretovať. V našom prípade vie, že má použiť sip schému. Ak by táto časť bola *mailto*, prekladač vie, že budú nasledovať meno používateľa @ samotná adresa [2].

SIP URI je definovaná ako

sip:user:password@host:port;uri-parameters?headers

Nie všetky časti sú však potrebné. Jediná povinná zložka je časť *host*, ktorá môže byť v tvare IP adresy alebo ako plne spôsobilé doménové meno. Naš prípad využíva zložky *user* a *host*.

Táto klasifikácia je výhodná z hľadiska implementácie. Aj keď staršie zariadenia nebudú rozumieť novo vytvorenej odpovedi, ktorá je správne klasifikovaná, stále budú môcť správne reagovať [3].

2.4 Všeobecné hlavičky

Niektoré polia protokolu SIP sa nachádzajú vo všetkých správach dialógu. Tieto polia nazývame všeobecné [4].

2.4.1 Pole Call-ID

Parameter Call-ID slúži na definovanie hovoru. Pomáha pri identifikácii hovoru. Takisto pri správach *INVITE* a *REGISTER*, ak správy prechádzajú cez proxy, ktoré duplikuje správy, alebo pri identifikácii hovoru počas konferencie. Tento parameter musí byť rovnaký pre všetky požiadavky a odpovede v rámci dialógu a mal by byť rovnaký pri každej registrácii používateľa. Tento parameter musí byť generovaný náhodne z dôvodu možného odcudzenia relácie treťou stranou.

2.4.2 Pole Cseq

Toto pole musí byť obsiahnuté vo všetkých požiadavkách a odpovediach. Týmto spôsobom vieme rozpoznať jednotlivé transakcie. Ako sme už spomínali, skladá sa z dvoch častí. Kladného čísla, ktoré sa na začiatku dialógu náhodne vyberie a mena metódy, ktorá začala transakciu požiadavkou. Každá nová transakcia SIP dialógu musí toto číslo zväčšiť o jedno. Jedinou výnimkou sa stávajú transakcie s metódami *ACK* a *CANCEL*. Tieto metódy číslo nemenia, kvôli ľahšej identifikácii, na ktorú transakciu boli odpoveďou. Druhá časť poľa *Cseq* sa mení vždy. V prípade retransmisie požiadavky sa číslo nemení. Ak sa požiadavka pri zaslaní protokolom UDP stratí a po určenom čase sa opätovne pošle, jej číslo zostáva rovnaké ako pri prvom pokuse.

2.4.3 Pole From

Toto pole je jedným z polí, ktoré identifikujú dialóg. Obsahuje voliteľné „zobrazené meno“, kontaktný údaj v tvare URI a položku *tag*. Hodnota celého poľa je vždy v rámci transakcie skopírovaná z požiadavky.

Volajúci môže svoju identitu skryť. V tomto prípade je odporúčané podľa RFC 3261, aby táto hodnota bola „*Anonymous*“. Druhá časť poľa je naďalej povinná, ale môže obsahovať ľubovoľné hodnoty, pričom však musí byť syntakticky správna.

Hodnota *tag* musí obsahovať ľubovoľný náhodný reťazec obsahujúci minimálne 32 bitov. Táto hodnota musí byť globálne unikátna, pretože sa stáva jednou z identifikátorov hovoru. V prípade rozdelenia hovoru proxy serverom, táto hodnota zostáva rovnaká pre všetky rozposlané požiadavky. Spôsob generovania tohto reťazca nie je definovaný a je ponechaný na každú implementáciu zvlášť.

2.4.4 Pole To

Pole tvorí identifikátor dialógu. Jednotlivé atribúty poľa sú podobné poľu *From*. Taktiež musí obsahovať URI a pole *Tag*. Tieto atribúty sa vytvárajú rovnako. Prvá časť sa preberie z prvej požiadavky v transakcii. Ak sa nachádza v prvej požiadavke, tak ju musí vygenerovať klient. Zapíše do nej cieľ svojej požiadavky. Atribút *Tag* je vygenerovaný rovnako ako v poli *From*, náhodne.

2.4.5 Pole Via

Toto pole môže byť v jednej požiadavke alebo odpovedi viackrát. Musí ho pridať každý server, ktorý preposiela SIP správy k cieľu (proxy). Vždy nové pole *Via* sa pridá nad všetky existujúce. Pomocou poľa *Via* je umožnené používateľom využívať protokol SIP, aj keď využívajú funkciu smerovača NAT (*Network Address Translation*). Toto pole môže obsahovať niekoľko atribútov. Keď klient vytvára požiadavku, musí toto pole obsiahnuť v správe. Musí obsahovať meno protokolu a jeho verziu, teda SIP a 2.0. Pole *Via* musí ďalej obsahovať parameter *branch*, ktorý v čase musí byť pre jedného klienta unikátny. Tento atribút spoľahlivo určuje požiadavku. Jeho hodnota je náhodná a ľubovoľná, musí však obsahovať začiatkový reťazec z9hG4bK, ktorým vieme určiť verziu protokolu. Staršie verzie protokolu SIP nemohli vygenerovať takýto reťazec. Ďalej obsahuje meno protokolu 4. vrstvy RM OSI modelu, IP adresu a port zariadenia, ktoré vytvorilo pole a parameter *rport* a *received*. Parameter *received* slúži na zaznamenanie IP adresy, z ktorej prišla správa, ak sa nezhoduje s hodnotou IP adresy v prijatom poli *Via*. Rovnako parameter *rport*, ktorý zaznamená port z ktorého správa prišla, v prípade, ak sa líši od hodnoty, ktorá bola zistená z poľa *Via*. Na základe týchto hodnôt vie server prekonať NAT a kontaktovať klienta.

Predstavme si situáciu, keď je klient za smerovačom s funkciou NAT *overload* (NAT-PT - *Network Address Translation - Protocol Translation*) a kontaktuje server správou *INVITE*, ktorú server prepošle volanému používateľovi. Klient musí vložiť do požiadavky pole *Via*. Toto bude obsahovať hodnoty

SIP/2.0/UDP 192.168.1.2:5067;branch=z9hG4bK6b1a9856;rport

Tieto hodnoty určujú IP adresu klienta a zdrojový port. Smerovač s funkciou NAT preloží aj ako IP adresu, tak aj zdrojový port. Proxy server správu prijme a zistí, že IP adresa a port prijatého IP paketu sa nezhodujú s hodnotami v prijatom poli *Via*. Do existujúceho poľa *Via* pridá atribút *received*, kam vloží IP adresu, z ktorej prijal správu, a atribút *rport*, kam vloží hodnotu portu, z ktorého bol paket prijatý. Následne vytvorí nové pole *Via*, ktoré zapíše nad už existujúce a vloží doň svoju IP adresu a port. Toto pole už ale musí obsahovať novú hodnotu *branch*, ktorá je rôzna od iných hodnôt *branch* v požiadavke. Takúto správu môže poslať požadovanému klientovi. Klient odpovie napríklad odpoveďou 200 OK, ktorou oznámi že prijíma hovor. Polia *Via* prepíše z požiadavky a celú správu pošle na adresu, z ktorej požiadavku prijal, teda na adresu servera. Ten správu prijme, skontroluje prvé pole *Via*, zistí, že ho vložil on a odstráni ho. Následne prečíta nasledujúce pole *Via*, a správu odošle na adresu a port, ktoré predtým vložil do atribútu *received* a *rport*. Takýto IP paket prijme smerovač volajúceho a podľa portu ho pošle na internú IP adresu, z ktorej bola požiadavka *INVITE* odoslaná.

2.4.6 Pole Content-Type

Označuje typ obsahu tela správy. Pre nás je zaujímavý typ *application/SDP*, ktorý hovorí, že v tele správy sa budú nachádzať informácie protokolu SDP, ktorý je popísaný nižšie. Toto pole je povinné, ak je obsah tela správy nenulový. Ďalším príkladom hodnoty poľa môže byť *text/html*.

2.4.7 Pole Content-Length

Určuje dĺžku tela správy v desiatkovej sústave v oktetoch bitov poslaných odosielateľom. V prípade nulového obsahu musí toto pole obsahovať číslo 0. Toto pole je povinné, ak sa používa spoľahlivý protokol ako TCP [1].

2.5 Odpovede protokolu SIP

Odpovede môžeme rozdeliť na dva typy: finálne a dočasné. Dočasné odpovede, narozdiel od finálnych, neukončujú transakcie a iba spresňujú a informujú stranu klienta o prebiehajúcich udalostiach. Odpovede sa delia do šiestich skupín. Každá skupina je reprezentovaná číslom (1xx, 2xx, 3xx, 4xx, 5xx, 6xx). Jednotlivé čísla reprezentujú konkrétnu odpoveď.

Konkrétne zaradenie odpovedí do skupín spolu s najdôležitejšími prípadmi sú:

- **1xx – informatívne**, *100 TRYING*, *180 RINGING*
- **2xx – oznamujúce úspešné vykonanie**, *200 OK*
- **3xx – presmerovanie**, *301 MOVED PERMANENTLY* (trvalo presmerované), *302 MOVED TEMPORARILY* (dočasne presmerované)
- **4xx – chyby na strane klienta**, *400 BAD REQUEST* (chybná požiadavka), *401 UNAUTHORIZED* (neautentifikované, zakázaná požiadavka), *404 NOT FOUND* (neidentifikovaný používateľ), *407 PROXY AUTHENTICATION REQUIRED* (potrebná autentifikácia na proxy serveri), *486 BUSY HERE* (používateľ nedosiahnuteľný)
- **5xx – chyby na strane servera**
- **6xx – globálne zlyhania** [1, 3]

2.6 SIP a nespoľahlivé protokoly

Pôvodné RFC 2543 žiadalo od implementácií použitie protokolu UDP. Dôvod na výber tohto protokolu je nižšia odozva a flexibilnejšia kontrola preposielania dát. Dnešné linky sú už menej náchylné na odozvu a tak sa väčším problémom stáva fragmentácia správ. Protokol SIP nemá definované žiadne mechanizmy na kontrolu fragmentácie, ani na vyrovnanie sa so stratou časti správy. Preto nové RFC 3261 vyžaduje implementáciu ako protokolu UDP, tak aj protokolu TCP a na komunikáciu odporúča protokol TCP. Všetky implementácie musia byť schopné prenášať najväčšie možné správy. V prípade UDP je to 65535 bajtov (vrátane IP a UDP hlavičiek) [RFC3261].

Transakcie v zmysle nespoľahlivých protokolov môžeme rozdeliť na *INVITE* a nie *INVITE* transakcie. V oboch prípadoch sa spoliehame na retransmisiu, keďže UDP protokol nemá nástroje na detekciu straty informácie.

2.6.1 Nie INVITE transakcie

Tieto správy sú charakteristické dvojcestnou výmenou a teda jedným potvrdením. V prípade, že zo strany servera (rozumie sa ako volaná strana) nedôjde k odpovedi na požiadavku do 500 ms (hodnoty časovačov v tejto kapitole sú pre zjednodušenie prevedené na ich štandardné hodnoty definované v RFC 3261) klient požiadavku prepošle. Ak naďalej neregistruje odpoveď, toto oneskorenie zdvojnásobí. Nasledujúca požiadavka sa prepošle po jednej sekunde. Takto pokračuje, až oneskorenie retransmisie dosiahne 4 sekundy. Klient sa bude takýmto spôsobom pokúšať poslať požiadavku po dobu 32 sekúnd. Ak je počas tejto doby prijatá nepovinná odpoveď, interval retransmisie sa ihneď posunie na 4 sekundy. Po dobe 32 sekúnd je relácia považovaná za stratenú. V prípade oznámenia od nižších vrstiev o strate IP konektivity (*ICMP unreachable paket*) je relácia ukončená okamžite.

Ak je stratená odpoveď zo strany servera, klient opätovne prepošle požiadavku, na ktorú server následne odpovie. Server odpovede nepreposiela.

Nepovinným odpovediam protokol SIP doručenie negarantuje. Keďže niektoré nepovinné odpovede sa stali v určitých implementáciách potrebné, do dokumentu RFC 3261 bola pridaná požiadavka *PRACK* (ak je potrebné aby počas čakania na odpoveď nebol audio kanál prázdny, *RINGING* odpoveď zabezpečí, aby opačná strana počas doby čakania vyzváňala).

2.6.2 INVITE transakcie

INVITE transakcia je trojcestnou výmenou. Je teda nutné potvrdiť potvrdenie. Ak chce používateľ nadviazať hovor, je prirodzené, že volaná strana neprijme hovor okamžite. Preto požiadavka *INVITE* narozdiel od iných požiadaviek, nedostane žiadnu informáciu o tom, že správa bola prijatá druhou stranou. Klient *INVITE* požiadavku prepošle po pol sekunde a neskôr túto dobu zdvojnásobuje. Až do doby, kedy prijme finálnu alebo nepovinnú odpoveď. Čakanie na odpoveď môže trvať až 180 sekúnd. Ak klient prijme nepovinnú odpoveď, požiadavku *INVITE* už nepreposiela. Ak klient zo strany servera, aj napriek zaslanej nepovinnej odpovede, pošle opäť požiadavku *INVITE*, server vie, že odpoveď sa stratila a nepovinnú odpoveď prepošle.

Prijatím hovoru alebo presmerovaním, volaná strana vyšle finálnu odpoveď 2xx alebo 3xx. V prípade nepotvrdenia požiadavkou *ACK* druhou stranou, server tieto odpovede prepošle za pol sekundy a následne túto dobu zdvojnásobuje. Túto retransmisiu ukončí iba požiadavka *BYE* a v prípade 3xx, 4xx, 5xx odpovedí to je požiadavka *CANCEL*. Tretí krok výmeny je požiadavka *ACK*. Ako sme už spomínali, jej hlavné polia sú totožné s *INVITE* požiadavkou. Pole *Cseq* má číselnú hodnotu taktiež rovnakú, ale meno metódy je *ACK*. Retransmisia požiadavky *ACK* je podobná nie *INVITE* požiadavkám. Ak server neprijme *ACK* požiadavku, znovu prepošle finálnu odpoveď. Takto sa klient dozvie, že jej požiadavka bola stratená.

Odpoveď *200OK* je jediná, ktorá musí byť potvrdená globálne, až ku klientovi. Ak je medzi volanými stranami proxy server, odpovede 3xx, 4xx, 5xx môžu byť potvrdené iba proxy serverom a klient sa o nich nemusí dozvedieť. Napríklad, keď volaná strana má viacero zariadení, ktoré zvonja postupne, teda nie všetky naraz, a prvé zariadenie odmietne hovor, toto odmietnutie potvrdí len proxy server. V prípade finálnej odpovede *200OK*, táto informácia musí byť preširoená až ku klientovi. Takýmto spôsobom sa o prijatí hovoru dozvedia všetky proxy servery medzi klientom a serverom.

V prípade spoľahlivých protokolov ako je TCP, sa správy nepreposielajú. Jedinou výnimkou sú finálne odpovede 2xx, ktoré musia byť preposlané aj protokolom ako je TCP. Ak by niektorý proxy server zmenil protokol na UDP, tak aj v takomto prípade môžeme zaručiť doručenie odpovede [1, 3, RFC2543].

2.7 Protokol SDP

Protokol SIP bol od začiatku navrhovaný ako jednoduchý a prispôsobivý protokol. Umožňuje teda nadviazať hovor s rôznou škálou parametrov a vlastností. Aby boli obe strany o týchto vlastnostiach informované, musia byť informovaní prostredníctvom transakcie. V našom príklade to bolo v rámci prvej transakcie. SIP protokol slúži len na nadviazanie relácie a nemá metódy na oboznámenie náprotivka o chcenom spôsobe komunikácie. Preto sa v transakcii spolu so

2 Protokoly SIP a SDP

SIP správy odosiľajú aj vnorené správy protokolu SDP (*Session Description Protocol*). Tento obsahuje všetky informácie na vhodné nastavenie RTP prenosu.

Protokol SDP je definovaný v RFC 2327. Protokol definuje nastavenia protokolu RTP ako port nad protokolom UDP, kódovanie obrazu a zvuku, základné informácie o relácii ako aj kontaktné informácie. Informácie v protokole nie sú špeciálne kódované a prenášajú sa vo forme `typ = hodnota`. Najdôležitejšie hodnoty sú

- **v** – verzia, hodnota je vždy 0
- **o** – vlastník, informácie o užívateľskom mene, ID relácie, verzia, typ siete, typ adresy, adresa
- **s** – názov relácie
- **c** – spojenie, typ siete, typ adresy, adresa
- **t** – čas, počas ktorého je relácia aktívna. Hodnoty sú dve čísla, medzi ktorými je relácia aktívna. Čas je reprezentovaný v sekundách vo formáte NTP
- **m** – opis média. Hodnoty v poradí znamenajú typ média, port, typ transportného protokolu, čísla určujúce typ kódovania v poradí v akom klient preferuje ich použitie. Číslo portu je vždy párne číslo. Nasledujúce nepárne číslo je číslo portu pre protokol RTCP (*Real-Time Control Protocol*).
- **a** – atribúty média

Príkladom informácie prenášanej SDP protokolom je napríklad:

```
v=0
o=Nokia-SIPUA 63390712757083500 63390712757083500 IN IP4
85.248.125.41
s=-
c=IN IP4 213.192.59.91
t=0 0
m=audio 57854 RTP/AVP 96 0 8 97 18 98 13
a=sendrecv
a=ptime:20
a=maxptime:200
a=fmtp:96 mode-change-neighbor=1
a=fmtp:18 annexb=no
a=fmtp:98 0-15
a=rtpmap:96 AMR/8000/1
a=rtpmap:0 PCMU/8000/1
a=rtpmap:8 PCMA/8000/1
a=rtpmap:97 iLBC/8000/1
a=rtpmap:18 G729/8000/1
```

2 Protokol SIP a protokol SDP

a=rtpmap:98 telephone-event/8000/1

a=rtpmap:13 CN/8000/1

3 IP Multimedia Subsystem

V posledných dvadsiatich rokoch mobilné a pevné komunikačné siete prešli veľkými zmenami. Vo svete mobilných sietí to boli najprv siete prvej generácie (1G), potom siete druhej generácie (2G) a dnes sú to už siete tretej generácie (3G), či (3.5G).

V pevných sieťach dominovali na prenos hlasového signálu *Public Switched Telephone Network* (PSTN) a *Integrated Services Digital Network* (ISDN) siete. V posledných rokoch sa záujem o Internet rapídne zvýšil a tým viac a viac používateľov začalo využívať výhody pripojenia technológie *Asymmetric Digital Subscriber Line* (ADSL).

Tento typ pripojenia naštartoval real-time komunikáciu, ako napr.: chatovacie aplikácie, online hry, či *Voice over IP* (VoIP). Aplikácie, ktoré ponúkajú služby zdieľaného prehliadania (*shared browsing*), sú vlastne peer-to-peer entitami. Schopnosť vytvoriť peer-to-peer spojenie v IP protokole je jednou z kľúčových súčastí týchto aplikácií. Táto forma v komunikácii presiahla možnosti *Plain Old Telephone Service* (POTS) sietí.

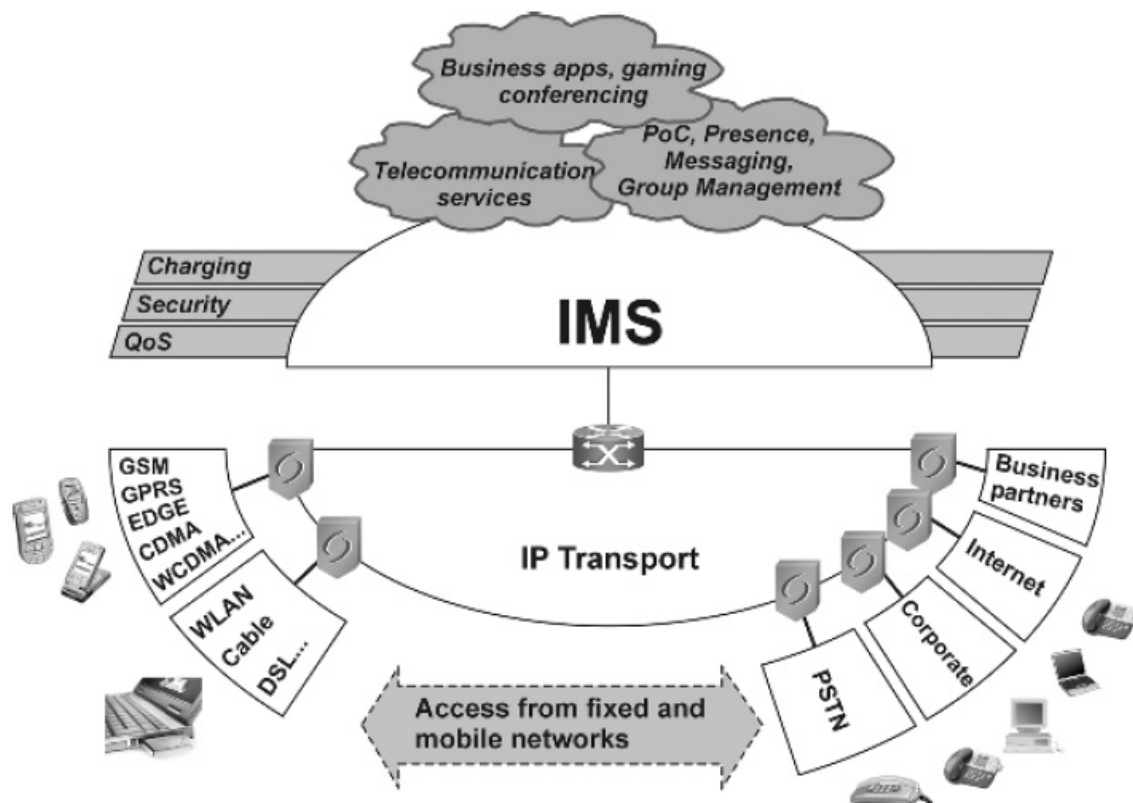
Aby aplikácie založené na protokole IP mohli komunikovať, museli mať vytvorený tento mechanizmus na spojenie s daným korešpondentom. V súčasnosti telefónne siete zaisťujú tento mechanizmus na vytvorenie spojenia. Pri vytáčaní *peer_a*, je vytvorené „ad-hoc“ spojenie medzi dvoma terminálmi prostredníctvom IP siete. Túto možnosť však ponúkajú len zariadenia poskytovateľov v Internete. A teda pri komunikácii s uzavretými systémami bolo potrebné vytvoriť globálny systém - IP Multimedia Subsystem (IMS). IMS dovolilo aplikáciám v IP sieťach vytvoriť *peer-to-peer* a *peer-to-content* spojenie ľahko a bezpečne.

Definícia podľa [5]:

„IMS je globálna, prístupovo nezávislá (*access-independent*) a postavená na štandarde IP spojenia architektúra kontrolných služieb, ktorá umožňuje používateľom používať rôzne typy multimediálnych služieb vďaka bežným IP protokolom.“

Vhodná integrácia hlasových a dátových služieb zvyšuje produktivitu a celkovú efektívnosť pri vývoji inovatívnych aplikácií integrujúcich hlas, dáta a multimédia, napr.: multimediálny chat, *push to talk* alebo konferenčný hovor.

Na obr. 3.1 je znázornená konvergovaná komunikačná sieť pre pevné a mobilné prostredia. Je to práve IMS, ktoré zaviedlo *multimedia session control* v paketovo prepínaných (*packet-switched*) doménach a v tom istom čase prinieslo funkcionality prepínania okruhov (*circuit-switched*) v paketovo prepínaných sieťach.



Obr. 3.1: IP Multimedia Subsystem [5]

3.1 Architektúra IMS

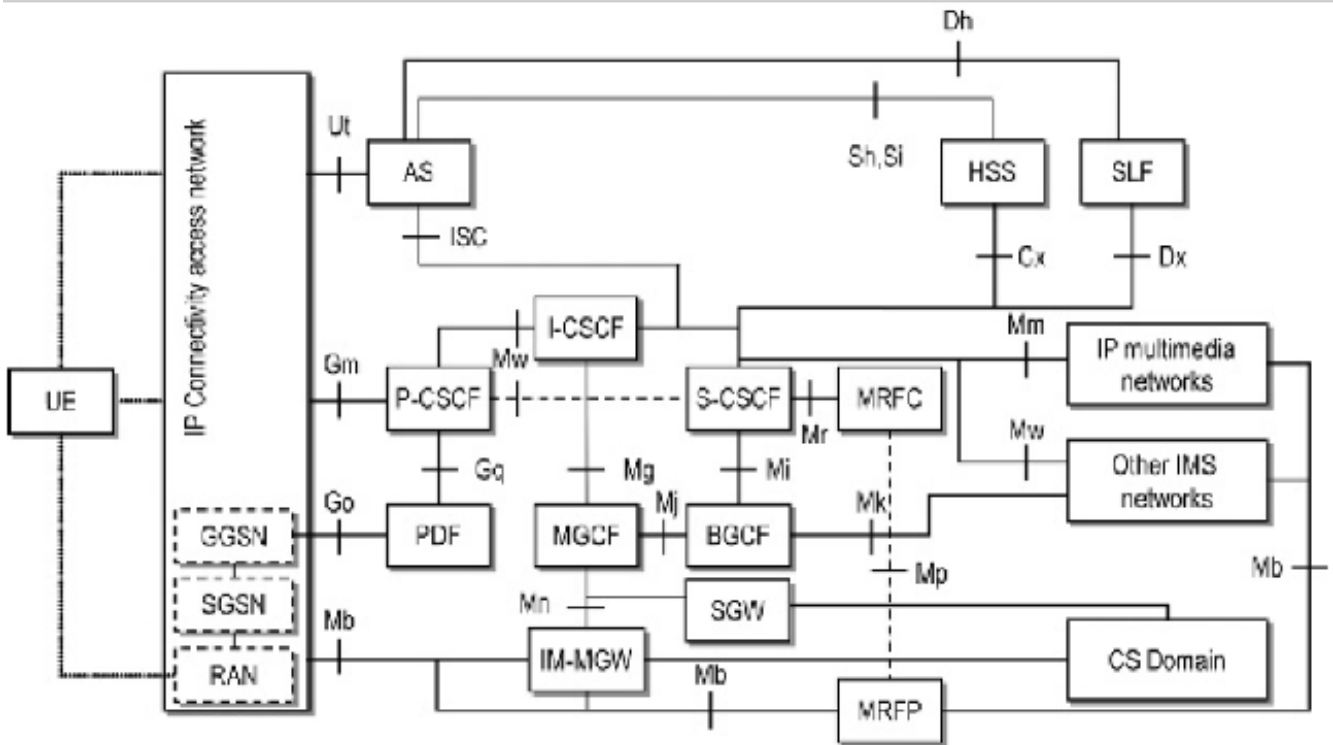
V tejto kapitole budú popísané jednotlivé entity a funkcie IMS. Treba poznamenať, že ohniskom IMS je kontrola relácií (*session control*). Keď sa pozrieme na 3GPP štandard pre IMS, nevidíme žiadne prepínače, či iné sieťové uzly ako súčasť IMS.

3.1.1 IMS entity a funkcie

IMS entity môžeme rozdeliť do približne šiestich kategórií [5]:

- *Call Session Control Function* (CSCFs)
- Databázy (HSS, SLF)
- Služby (AS, MRFC, MRFP)
- Vnútorne funkcie (BGCF, MGCF, IMS-MGW, SGW)
- Podporné funkcie (PDF, SEG, THIG)

Jednotlivé prepojenie týchto entít je znázornené na obr. 3.2.



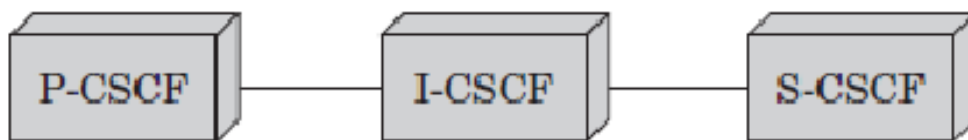
Obr. 3.2: Architektúra IMS

3.1.2 Call Session Control Function (CSCF)

Call session control je primárna funkcia jadra siete. Táto funkcionálnosť je distribuovaná v celej sieti, aby zabezpečovala efektívnosť a škálovateľnosť. Existujú tri typy *Call Session Control* funkcií:

- *Proxy-CSCF* (P-CSCF),
- *Serving-CSCF* (S-CSCF)
- *Interrogating-CSCF* (I-CSCF)

Každá funkcia CSCF má vlastnú úlohu, ale spoločne pracujú pri registrácii a zriadení relácie a pri SIP smerovaní. Navyše, všetky funkcie sú schopné posielať údaje *offline*. Rozdiel medzi týmito funkciami je znázornený na obr. 3.3 spočívajúci v ich rozdielnom účele a procedúrach, ktoré vykonávajú [6].



Obr. 3.3: Funkcie IMS

Proxy CSCF (P-CSCF)

Proxy Call Session Control funkcia (P-CSCF) je prvým prístupovým bodom k SIP doméne z perspektívy kontroly relácie. Cez P-CSCF prechádzajú SIP správy. Prvá komunikácia je registrácia

IP adresy zariadenia. Ak zariadenie chce komunikovať s inými zariadeniami, musí najprv s daným zariadením zahájiť reláciu. Táto relácia je najprv vytvorená cez P-CSCF. Skôr ako je daná správa poslaná ďalej C-SCF, P-CSCF pridá do hlavičky vlastné parametre.

Keďže SIP protokol je textovo orientovaný, 3GPP kvôli zrýchleniu inicializácie relácie poverilo P-CSCF kompresiou protokolu SIP. Ďalšie úlohy pre P-CSCF sú napríklad: IPsec bezpečnosť, interakcia s *Policy Decision Function* (PDF) a detegovanie mimoriadnych relácií (*Emergency Session Detection*).

P-CSCF je vstupným bodom do IMS, zohráva z hľadiska bezpečnosti dôležitú úlohu.

V P-CSCF je prítomné *Policy Decision Function* (PDF), ktoré rozhoduje ako sa správať v špecifických scenároch. PDF dovoľuje operátorom vytvoriť pravidlá pre prístup do siete. PDF kontroluje *Policy Enforcement Function* (PEF) na nosnej sieti. To povoľuje operátorom kontrolovať tok paketov na nosnej sieti v súlade s cieľovou a zdrojovou adresou a právami.

P-CSCF takisto preveruje smerovanie. Kontroluje, že či prijaté smerovanie v SIP žiadosti alebo odpovedi je to isté smerovanie, ktoré bolo identifikované, keď sa dané zariadenie zaregistrovalo do siete. Ak smerovacie hlavičky neobsahujú adresy zhodné s uloženými adresami počas registrácie v P-CSCF, znamená to, že boli zmenené.

P-CSCF si uchováva tieto informácie: adresu zariadenia (IP adresu) a verejné a privátne informácie používateľov.

V P-CSCF sú vedené informácie o stave relácie počas celého spojenia. V prípade, že zariadenie stratí IP konektivitu, P-CSCF o tom notifikuje IMS prostredníctvom správy *CANCEL* všetkým účastníkom danej relácie.

Keď to všetko zhrnieme, môžeme povedať, že P-CSCF je akási brána do IMS. Je zodpovedná za to, že zariadenie, ktoré pristupuje do siete, je registrované a má povolený prístup do IMS, avšak nezabezpečuje autentifikáciu a autorizáciu [6].

Interrogating CSCF (I-CSCF)

Zatiaľ čo P-CSCF je vstupným bodom do IMS, I-CSCF slúži ako brána do každej individuálnej IMS siete. I-CSCF pomáha chrániť S-CSCF a HSS pred neautorizovaným prístupom z iných sietí. Keď S-CSCF prepošle požiadavku alebo odpoveď do inej siete, správa je ďalej preposlaná do I-CSCF, ktoré danú správu prepošle ďalej do cieľovej siete.

Dôležitou funkciou I-CSCF je priradenie S-CSCF. S-CSCF je priradené v závislosti od schopnosti a samozrejme politiky poskytovateľa služieb.

Jednotlivé úlohy I-CSCF by sme mohli zhrnúť nasledovne [5]:

- Získanie mena nasledujúceho uzla (S-CSCF alebo aplikačný server) z *Home Subscriber Server* (HSS)
- Priradenie S-CSCF na základe prijatých schopností z HSS
- Priradenie prichádzajúcich smerovacích požiadaviek k S-CSCF alebo aplikačnému serveru
- Poskytovanie *Topology Hiding Inter-network Gateway* (THIG) funkcionality

Serving CSCF (S-CSCF)

S-CSCF je jadrom IMS. Kontroluje všetky aspekty odberateľských služieb, pričom eviduje stav každej relácie, ktorá bola inicializovaná. S-CSCF kontroluje správy a doručenie obsahu.

Takisto sprostredkúva informácie o stave registrácie iným aplikáciám a udržiava kontrolu počas celej doby, ako je dané zariadenie registrované.

Zo SIP perspektívy je S-CSCF registrátor, ktorý je schopný autentifikácie odberateľov, ktorí sa pokúšajú registrovať.

S-CSCF udržiava nasledovné informácie o zaregistrovanom zariadení [5]:

- HSS adresu (*HSS address*)
- Používateľský profil (*User profile*)
- P-CSCF adresu (*P-CSCF address*)
- P-CSCF doménu (*P-CSCF domain*)
- Verejnú používateľovu identitu (*Public user identity*)
- Privátnu používateľovu identitu (*Private user identity*)
- IP adresu zariadenia (*Device IP address*)

S-CSCF sprístupňuje služby poskytovaním prístupu k rôznym aplikačným serverom (*Application Servers - ASs*) v sieti. To znamená, že S-CSCF potrebuje poznať, že ku ktorej službe má odberateľ povolený prístup a samozrejme adresu servera, ktorý danú službu poskytuje. S-CSCF pristupuje k HSS, aby získalo používateľov profil, ktorý obsahuje aj profil služieb.

Takisto aj konvertovanie adresy je rola S-CSCF. Odkedy SIP smeruje na základe SIP URI, každé TEL URI musí byť preložené do SIP URI. Podobne je to aj so smerovaním z IMS do PSTN. S-CSCF pristupuje do ENUM/DNS pre konvertovanie adresy do SIP URI na preposielanie správ adresátovi.

ENUM/DNS aplikácie sú umiestnené na tom istom serveri alebo môžu byť ako samostatné ENUM funkcie.

S-CSCF pracuje ako statické proxy a musí si udržiavať stav všetkých registrácií a relácií, ktoré má pod kontrolou.

Jednoducho povedané S-CSCF je jadro IMS. Je to podstata alebo bod siete, ktorý poskytuje operátorom kontrolu doručenia služieb a všetkých relácií.

3.1.3 Databázy

IMS pracuje s dvoma databázami: *Home Subscriber Server (HSS)* a *Subscription Locator Function (SLF)* [6].

Home Subscriber Server (HSS)

HSS obsahuje hlavné údaje o všetkých používateľoch a údaje súvisiace so službami IMS. Sú to údaje ako: identita používateľa (verejná a privátna), informácie o registrácii, prístupové parametre a informácie o spúšťaní služieb [3GPP TS 23.002].

Hlavnou funkcionalitou HSS je šifrovanie a autorizovanie každej správy. Keď sa odberateľské zariadenie zaregistruje do siete, priradené S-CSCF vyzve zariadenie na zadanie správnych údajov, ktoré sú uložené v HSS [6].

Subscription Locator Function (SLF)

Táto funkcionálnosť môže byť vstavaná do inej databázy alebo môže fungovať ako samostatný server. Hlavným účelom tejto funkcie je lokalizovanie HSS a S-CSCF, ktoré je priradené danému odberateľovi. Je to indexovanie a mapovanie používateľskej identity k S-CSCF/HSS v závislosti od registrácie [6].

3.1.4 Služby

Jednotlivé služby môžeme rozdeliť na tri kategórie: *Multimedia Resource Function Controller* (MRFC), *Multimedia Resource Function Processor* (MRFP) a Aplikáčnne servery (AS) [5].

Služby MRFC a MRFP

MRFC a MRFP spolu obsluhujú nosné služby ako konferenciu, správy k používateľovi alebo nosné kódovanie v IMS architektúre.

MRFC má za úlohu udržiavať SIP komunikáciu od S-CSCF a k S-CSCF a kontroluje MRFP. MRFP zabezpečuje nasledovné funkcie [5]:

- Spracovanie SIP relácií prijatých z IMS
- Schopnosť zahájenia SIP žiadostí
- Schopnosť posielania informácií o konte

Aplikačné servery

Aplikačný server - AS má viac úloh v IMS, hoci nie je časťou IMS jadra. Existuje veľa možností využitia AS ako napr. aplikácie, doručovacie služby, doručenie správ, prezentačný server či video konferencia. Jednotlivé služby, ktoré poskytuje AS, sú identifikované identifikátorom (adresou).

AS vie vytvoriť SIP dialóg, čo znamená, že sa vie tváriť ako používateľský klient pre SIP [6].

3.1.5 Vnútorne funkcie

V literatúre sú spomínané štyri vnútorné funkcie, ktoré sa starajú o signalizáciu medzi IMS a *Circuit Switched Core Network* (CS CN).

Breakout Gateway Control Function (BGCF) je brána k sieťam iných poskytovateľov.

MGCF (*Media Gateway Control Function*) je pripojená do PSTN domény a zabezpečuje funkcie brány medzi SS7 (*Signaling System 7*) a IP/SIP stranou.

Keď SIP požiadavka dorazila do MGCF, v MGCF musela byť vykonaná konverzia medzi SIP protokolom a *ISDN User Part* (ISUP) alebo *Bearer Independent Call Control* (BICC). Táto správa bola poslaná a konvertovaná *Signaling Gateway* (SGW) do CS CN [5].

3.1.6 Podporné funkcie

Policy Decision Function (PDF) je zodpovedné za politiku rozhodovania sa na základe informácií získaných z relácie a multimediálne orientovaného obsahu prichádzajúce z P-CSCF.

Security Gateway (SEG) má za úlohu chrániť prevádzku medzi bezpečnostnými doménami. V IMS je prevádzka smerovaná cez SEG, špeciálne vtedy, keď zdrojová a cieľová doména sú v rôznych bezpečnostných doménach [5].

Topology Hiding Inter-network Gateway (THIG) je používaná na ukrytie konfigurácie, kapacity a topológie siete od siete, ktorá sa nachádza na druhej strane operátora. Keď chce operátor využiť ukrytie týchto funkcií, musíme umiestniť THIG na ceste smerovania správ pri prijímaní alebo odosielaní z IMS sietí.

4 Optimalizácia siete

Návrh siete so všetkými prostriedkami v štandardnej a nezmenenej konfigurácii funguje a funguje dobre. Avšak len do chvíle, keď sa objaví problém na sieti, alebo zmena v topológii siete. Čas, do ktorého sa sieť „spamätá“, je v implicitnej konfigurácii dlhý. Naším cieľom je optimalizovať sieť predovšetkým na sieťovej vrstve tak, aby sme dosiahli zlepšenie parametrov, skrátenie času konvergenzie, kedy je sieť v nestabilnom stave a zároveň minimalizovali negatívne dôsledky optimalizácie.

Pri optimalizácii sa stretávame s pojmom konvergencia. Konvergencia je schopnosť a rýchlosť spolupracujúcich zariadení v sieti aktualizovať svoje údaje na nové hodnoty po zmene pomerov v sieti. Konvergencia je proces. Stav, kedy je konvergencia ukončená, budeme nazývať skonvergovaným stavom. Konvergencia je kľúčový aspekt v moderných sieťach. Pri konvergencii sledujeme najmä čas jej trvania. Čas konvergenzie je trvanie konvergenzie od vzniku zmeny, jej detekcie, až po záverečnú dohodu a aktualizáciu na zariadeniach. Je to čas, ktorého trvanie vyjadruje porušený skonvergovaný stav. Rýchlejšia konvergencia znamená kratšiu dobu nedostupnosti siete. Je to práve čas, ktorý sa stal kritickým faktorom dobre fungujúcej siete. Neexistuje okamžitá konvergencia, pretože tento proces vždy trvá nejaký čas.

4.1 Topológia

Sieť je nutné začať optimalizovať už od fyzickej vrstvy, teda od návrhu siete. Ak sieť nie je možné na fyzickej vrstve nijakým spôsobom zmeniť, alebo len čiastočne, nepredstavuje to prekážku pre nadväzujúcu optimalizáciu na vyšších vrstvách.

Ako Filip Burda vo svojej bakalárskej práci [7] zistil, je veľmi ťažké rozhodnúť, ktorý typ topológie je optimálny pre rýchlu konvergenciu, pretože každá má svoje nedostatky aj z pohľadu konvergenzie. Všeobecne, zbernica je dobrá na rozšírenie informácie, no vďaka kolíziám sa môže stať absolútne nevhodnou voľbou. Pri topológiach založených na kruhu sa spoliehame, že nedôjde k výpadku v rámci kruhu. Výpadok automaticky môže znamenať vyradenie časti siete z prevádzky. Pri hviezde sa stáva kritickým uzlom už len centrálny rozbočovač. Navyše tým, že uzly sú vzájomne napojené na jeden centrálny uzol, konvergencia je veľmi rýchla. Z tohto pohľadu nám najlepšie vychádza práve hviezda, ktorá je dobrým kompromisom stability a konvergenzie.

Ideálnejšie riešenia však ponúkajú hybridné topológie. Veľmi zaujímavou voľbou je čiastočný *mesh*, ktorý je odolný aj voči výpadkom, aj poskytuje redundanciu a aj rýchlu konvergenciu. Táto voľba kombinovaná so stromovou topológiou je v praxi dobre realizovateľná a stále z pohľadu konvergenzie zaujímavá. Čiastočný *mesh* znamená prepojenie zariadení nie každého s každým, ale každého s niekoľkými. Stromová topológia je v podstate hierarchický návrh siete s postupným rozvetvovaním sa po istú úroveň.

Vhodný výber topológie je však len prvým krokom k naplneniu cieľa rýchlo konvergujúcej siete. V nasledujúcej časti sa pozrieme na rôzne typy zariadení, ich výber a príslušenstvo [8].

4.2 Zariadenia

Ak by sme chceli prepojiť každý koncový uzol so všetkými ostatnými koncovými uzlami po svete, asi by z našej planéty bolo len kľbko prepletených káblov. Na vzájomnú komunikáciu zariadení sa používajú rôzne zariadenia, ktoré slúžia ako medziuzly v komunikácii. Podľa toho, ako dokážu k dátovému toku pristupovať a narábať s ním, rozdeľujeme zariadenia na opakovače, rozbočovače, mosty, prepínače a smerovače. Samozrejme zariadení je na výber omnoho omnoho viac, no tieto sú základom siete. Z pohľadu poskytovateľa služieb sú zaujímavé smerovače a prepínače na tretej vrstve, ktoré popíšeme.

Smerovače pracujú na sieťovej vrstve a dáta smerujú. Dáta sú posielané väčšinou deterministicky. Vzájomné informovanie sa smerovačov zabezpečuje dosiahnuteľnosť všetkých častí celosvetovej siete.

Smerovače sú z vyššie spomenutých zariadení najviac inteligentné, avšak spôsobujú aj najväčšie oneskorenie, čo predlžuje čas konvergencie. Problém s prepínačmi je ten, že môže existovať len práve jedna cesta medzi dvomi akýmikoľvek prepínačmi. Toto obmedzenie predchádza vzniku topologických slučiek, ktoré by vyradili sieť z prevádzky. Čas oneskorenia je o niečo menší, ale práve toto obmedzenie naopak spomaľuje rozšírenie správy o zmene do siete, čo spomaľuje konvergenciu. Pokiaľ to je možné, mali by byť vybrané predovšetkým prepínače na tretej vrstve. Tieto zariadenia sú aj prepínačmi aj smerovačmi zároveň. Výhodou je veľký počet portov, typický pre prepínače a schopnosť nielen prepínania, ale aj smerovania v jednom zariadení. Avšak smerovanie narozdiel od smerovačov je realizované hardvérovo.

Pri výbere zariadenia zohľadňujeme platformu, vhodnosť pre našu sieť a jeho využiteľnosť. Neoddeliteľnou súčasťou sú aj schopnosti jeho operačného systému. Možnosti, ktoré sú nám ponúkané, sú pre nás dôležité. Postupná evolúcia v sieťach a vylepšenia v implementácii protokolov sa zavádzajú aj do operačných systémov zariadení. Odporúčame preto pred začiatkom optimalizácie nainštalovať najnovší operačný systém. Operačný systém (OS) sa napríklad na Cisco zariadeniach nazýva IOS (*Internetwork Operating System*). Na zariadeniach od firmy Juniper to je JUNOS. V práci sa budeme orientovať na Cisco zariadenia s IOS verziami 12.3 a 12.4 pre smerovače. Staršie verzie nemusia podporovať všetky techniky rýchlej konvergencie, ktoré popíšeme a nemusia byť možná optimalizácia časov na dosiahnutie konvergencie pod jednu sekundu pre daný smerovací protokol.

V ďalších kapitolách sa zameriame len na zbernicovú topológiu a hviezdu. Základným prenosovým „médiom“ bude *ethernet*, keďže je v súčasnosti najpoužívanejší a najrozšírenejší. Určite by bolo zaujímavé sa pozrieť aj na iné média, ale berieme taktiež ohľad na laboratórne podmienky.

Na linkovej vrstve máme viacero prostriedkov na zlepšenie konvergencie. Keďže sa v tomto projekte neuvažuje o linkovej vrstve vo vnútri siete u poskytovateľa služieb, nebudeme sa ani my venovať tejto optimalizácii. Podrobnejšie o tejto optimalizácii je možné nájsť v bakalárskej práci Filipa Burdu [7] a v jeho vedeckej publikácii [9].

Sieťová vrstva tvorí podstatnú zložku siete a budeme sa jej patrične venovať na nasledujúcich stranách.

5 Smerovanie v TCP/IP

Hlavnou úlohou sieťovej vrstvy je zabezpečiť najlepšie možné doručenie paketu od odosielateľa až k príjemcovi.

Zariadenia, ktoré implementujú túto funkcionálnu sieť sú smerovače. Tieto zariadenia musia urobiť rozhodnutia, cez ktoré rozhranie majú odoslať pakety tak, aby sa optimálne dostali k svojmu cieľu.

Pri tomto procese je nutné zabezpečiť, aby zariadenia, ktoré poskytujú takúto funkcionálnu, mohli so sebou komunikovať a aby sa mohli správať ako jeden konzistentný celok, ktorý si navzájom dobre rozumie. Týmto by bolo možné dosiahnuť, aby si vzájomne aktualizovali a spravovali svoje tabuľky. K tomuto účelu je možné využiť smerovacie protokoly. Tieto protokoly poskytujú algoritmy na ukladanie smerovacích informácií.

5.1 Klasifikácia smerovania

Smerovanie je proces výberu ciest v sieti medzi uzlami siete. Rozdeľujeme ho na dve základné skupiny: statické smerovanie a dynamické smerovanie.

5.1.1 Statické smerovanie

Statické smerovanie (*Static Routing*) je smerovanie, ktoré sa konfiguruje na každom smerovači individuálne, podľa návrhu topológie a rozloženia smerovačov v sieti. Pri akejkoľvek zmene topológie nastáva potreba manuálne prekonfigurovať zariadenia.

Tým, že si smerovače nevymieňajú informácie o smerovaní, neprichádza k výmene žiadnych obslužných paketov, ktoré následne nespotrebovávajú vyžadovanú šírku pásma.

Využitie statického smerovania sa v súčasnosti ukazuje výhodné v ich konfigurácii ciest v prípade predvolenej brány (*default gateway*). Je to v prípadoch, kedy je vhodné, aby administrátor po zvážení predvolil cestu, ktorou bude smerovač posilať pakety pokiaľ nenájde záznam v smerovacej tabuľke.

Ďalšie využitie statických ciest je v listových oblastiach (*Stub Networks*). Pri týchto oblastiach nedochádza k distribúcii paketov medzi viacerými oblasťami alebo autonómnymi systémami. Ak existuje len jeden východ z oblasti, je vhodné predvoliť cestu, ktorá je výstupnou z oblasti. Tieto oblasti budú popisované podrobnejšie pri protokole OSPF.

Medzi najväčšiu nevýhodu statického smerovania patrí nízka flexibilita. Čím viac sa topológia rozrastá, tým je väčšia náročnosť na správu takejto siete. To je dôvod, prečo sa statické smerovanie vo všeobecnosti považuje za nie veľmi vhodné. Obzvlášť nie je vhodné pre tzv. „živé siete“, v ktorých často dochádza k zmenám topológie.

5.1.2 Dynamické smerovanie

Pri statickom smerovaní akékoľvek rozšírenie siete znamená jej väčšiu zložitosť a náročnosť na údržbu. V dynamickom smerovaní tento problém do značnej miery odpadá.

Dynamické smerovanie, tiež známe ako adaptívne dynamické smerovanie (*Adaptive Routing*), popisuje schopnosť smerovačov dynamicky určovať najlepšie cesty v závislosti od zmeny topológie.

Za pomoci adaptívneho smerovania je možné odkloniť premávku kvôli výpadku uzla, ktorým práve prechádza optimálna trasa pri dostatočnej redundancii. V prípade výpadku uzla je potrebné vykonať nový prepočet zahrňujúci zmenu topológie.

Použitie adaptívneho smerovania nesie so sebou niekoľko negatívnych faktov. V prvom rade vyžaduje istú šírku pásma (*bandwidth*) a spotrebu procesorového času. Po získaní vhodných aktualizácií protokol príslušným algoritmom vypočíta, ktoré cesty sú najlepšie a budú použité na smerovanie, teda budú zahrnuté v smerovacej tabuľke (*routing table*).

Možnosť použiť adaptívne smerovanie v sieťach je významnejšie, ako spotreba časti šírky pásma na režiu, alebo náročnosti na procesor. Riešenie v podobe adaptívneho smerovania je oveľa flexibilnejšie.

5.2 Klasifikácia smerovacích algoritmov a protokolov

Smerovací protokol je protokol, ktorý špecifikuje spôsob komunikácie smerovačov medzi sebou. Smerovací algoritmus je súbor inštrukcií prináležiaci výpočtu najlepšej cesty smerovacieho protokolu. Samotné smerovacie protokoly môžeme rozdeliť podľa niekoľkých aspektov.

5.2.1 Aspekt rozsahu použitia

Aspekt tkvie v rozsahu použitia daného protokolu. Protokoly, ktoré sa využívajú na smerovanie prevádzky medzi viacerými autonómnymi systémami patria do skupiny ERP (*Exterior Routing Protocol*). Dnešné protokoly, ktoré sa používajú na smerovanie v rámci viacerých autonómnych systémov sú EGP (*Exterior Gateway Protocol*), BGP (*Border Gateway Protocol*) a CSPF (*Constrained Shortest Path First*).

Protokol EGP patrí do skupiny DV (*Distance Vector*) protokolov a je to predchodca protokolu BGP. Na svoje použitie vyžaduje stromovú topológiu. Dnes sa už nepoužíva.

Pre sieť Internet bol postupne okolo roku 1994 nahradený protokolom BGP. Súčasná verzia 4 je špecifikovaná v [RFC4271] z roku 2006, kde bolo pridané najmä CIDR (*Classless Inter-Domain Routing*) a sumarizácia, ktorá je dôležitá kvôli zníženiu veľkosti smerovacích tabuliek na smerovačoch. Podrobnejšie sa protokolu BGP budeme venovať v samostatnej časti.

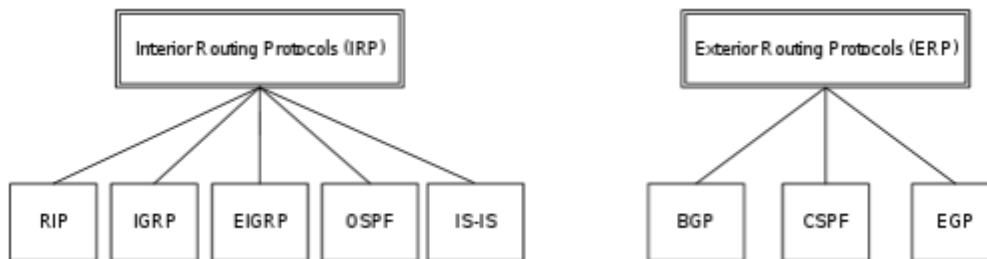
Protokol CSPF je rozšírený v MPLS (*Multiprotocol Label Switching*) sieťach. Je súčasťou CBR (*Constraint Based Routing*). Najlepšia cesta je počítaná rovnakým algoritmom ako v OSPF a IS-IS.

Iná situácia nastáva pri potrebe použitia protokolu navrhnutého na smerovanie v rámci jedného autonómneho systému. Tieto protokoly patria do skupiny IRP (*Interior Routing Protocol*). Hlavný aspekt delenia tejto skupiny protokolov tkvie vo výpočte optimálnej cesty.

Niekedy si želáme rýchlu konfiguráciu siete s jednoduchou konfiguráciou zariadení a ich údržby aj za cenu menšej efektivity. V prípade sietí, ktoré podliehajú veľkej záťaži (vzhľadom na ich kapacitu) nám zvyčajne záleží na efektívite prenášaných údajov, teda optimálnosti výsledkov výpočtov optimálnych ciest. Túto skupinu smerovacích protokolov tvoria RIP verzie 1 (*Routing Information Protocol*), RIP verzie 2, IGRP (*Interior Gateway Routing Protocol*), EIGRP (*Enhanced*

Interior Gateway Routing Protocol), OSPF (*Open Shortest Path First*), IS-IS (*Intermediate System to Intermediate System*) a Integrated IS-IS. Tieto protokoly sú detailnejšie popísané v nasledujúcich častiach.

Obr. 5.1 ukazuje rozdelenie hlavných protokolov do skupín podľa príslušnosti v skupine IRP alebo ERP.

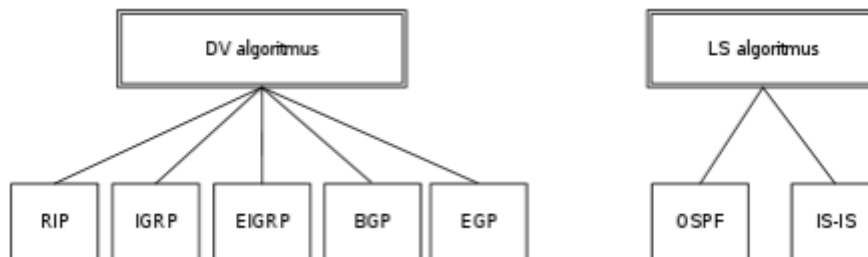


Obr. 5.1: Klasifikácia protokolov na základe rozsahu použitia

5.2.2 Aspekt smerovacieho algoritmu

Tu sa stretávame s dvoma možnosťami a s prípadnými modifikáciami. Prvú skupinu tvoria protokoly založené na algoritme *Distance-Vector* (ďalej DV). Druhú skupinu tvoria protokoly založené na algoritme *Link-State* (ďalej LS), ktorý poskytuje efektívnejšie možnosti správy výmeny smerovacích informácií za cenu relatívne vyšších nárokov oproti DV algoritmu. Tieto algoritmy budú popisované dôkladnejšie ďalej v tejto práci.

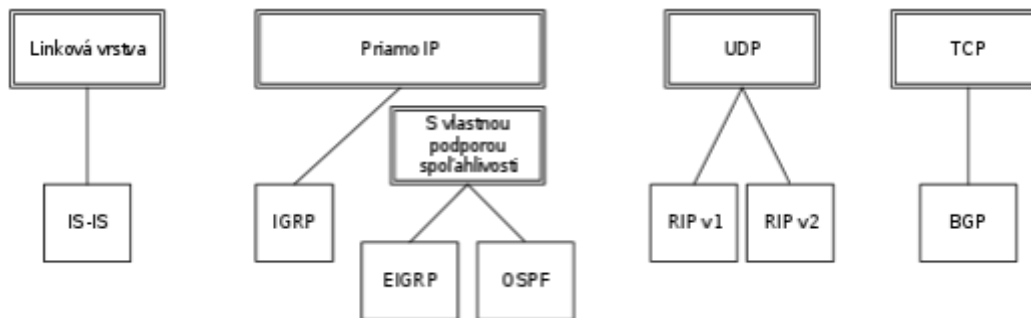
Obr. 5.2 rozdeľuje protokoly na základe ich príslušnosti k danému smerovaciemu algoritmu.



Obr. 5.2: Rozdelenie smerovacích protokolov podľa smerovacieho algoritmu

5.2.3 Aspekt transportného mechanizmu

Aspekt delenia smerovacích protokolov je založený na základe transportného mechanizmu. Od typu transportného mechanizmu závisí, akú spoľahlivosť pri prenose paketov sieťou dosiahneme. Z obrázku 5.3 je vidieť, že protokol IS-IS používa len linkovú vrstvu, IGRP používa priamo IP protokol, ktorý sám o sebe nie je spoľahlivý bez patričnej nadstavby (napr. TCP). OSPF a EIGRP si sami zaisťujú spoľahlivosť nad protokolom IP, ktorý používajú [10]. RIP verzie 1 aj 2 používa IP s nadstavbou UDP. Pozoruhodné je riešenie protokolu BGP, ktorý používa priamo TCP [RFC793] nad protokolom IP. Obr. 5.3 rozdeľuje protokoly na základe ich transportného mechanizmu.



Obr. 5.3: Rozdelenie smerovacích protokolov na základe ich transportného mechanizmu

5.3 Protokoly s DV algoritmom

Protokoly typu DV používajú ako základ Bellman-Fordov algoritmus. Tento algoritmus sa používa na nájdenie najkratšej cesty medzi dvoma uzlami siete. Algoritmus funguje aj pre negatívne hrany a vytvára kompletnú smerovaciu tabuľku.

Smerovače smerujúce cez DV protokoly si nevytvárajú komplexnú topológiu siete, nevytvárajú si nad ňou nadhľad. Protokolom DV sa nebudeme bližšie venovať, keďže nepodporujú nosnú technológiu tejto práce a to MPLS TE. Bližšie si popíšeme len externý smerovací protokol BGP.

5.3.1 Protokol BGP

Protokol BGP (*Border Gateway Protocol*) bol navrhnutý pre smerovanie medzi viacerými autonómnymi systémami (*autonomous systems*). Napriek svojej robustnosti a veľkej prispôsobivosti používa algoritmus DV. Na prenos informácií sa používa protokol TCP. Aktualizácie sú prírastkové (*incremental*) a odistené (*triggered*) bez priamej potreby ich opakovať vo vopred stanovenom intervale. Výber vstupov metriky pri BGP možno ladiť širokou paletou prostriedkov, ktoré budeme opisovať nižšie.

Z hľadiska rýchlosti konvergencie, je BGP bezpochyby najpomalší smerovací protokol v oblasti aktualizácie stavu topológie. Protokol BGP nebol navrhnutý na rýchlu konvergenciu, ale na stabilitu a správu veľkého množstva sietí s veľmi veľkým počtom ciest a smerovacích tabuliek.

V rozsiahlych sieťach WAN je bežné, že prakticky každú sekundu môže nastať zmena v sieti, teda zmena topológie, ktorú je treba oznamovať všetkým BGP uzlom v sieti. Pokiaľ by BGP konvergoval tak rýchlo ako protokoly primárne používané v sieťach LAN, nastalo by značné zahltenie celej siete WAN oznamovacími paketmi, čo by výrazne spomalilo celú sieť.

BGP rozpoznáva dva typy smerovania. Smerovanie v rovnakom autonómnom systéme sa označuje ako IBGP (*Internal Border Gateway Protocol*) a smerovanie medzi viacerými autonómnymi systémami sa nazýva EBGP (*External Border Gateway Protocol*). Toto rozčlenenie spôsobuje problémy so spätnou synchronizáciou pre zistenie optimálnych ciest, ktoré sa zapisujú do smerovacej tabuľky.

Smerovanie v BGP má svoje pevné pravidlá. Jedno z týchto pravidiel hovorí, že autonómny systém v ktorom BGP smeruje, nebude nikdy nútiť iný autonómny systém smerovať určeným

spôsobom. Teda môžeme určiť, akým spôsobom sa bude smerovať prevádzka v našom autonómnom systéme, ale nebudeme určovať dianie za jeho hranicou, aj keď to je možné.

Atribúty BGP

Keby sme do správania BGP nezasahovali v oblasti výberu najlepšej cesty, správal by sa podobne, ako protokol RIPv1. Spočítal by, koľko smerovačov je medzi ním a cieľovou sieťou a podľa toho by si vyberal cesty.

Na zmenu tohto implicitného správania BGP sa používajú atribúty (*attributes*).

Atribúty delíme z viacerých prístupov. Prvý prístup je delenie z hľadiska podpory výrobcami na dobre známe (*well-known*) a voliteľné (*optional*). Ďalej ich môžeme deliť podľa toho, či majú byť zahrnuté v aktualizáciách na povinné (*mandatory*) a popisné (*discretionary*). Niektoré aktualizácie môžu byť prenášané od smerovača na smerovač, podľa ktorých atribúty delíme na tranzitívne (*transitive*) a na netranzitívne (*non-transitive*). Pri tranzitívnom režime, aj pokiaľ aktualizáciu nerozpozna, posieľa ju ďalej s predpokladom, že bola určená pre iný BGP smerovač, ktorý ju rozpozná. Netranzitívne aktualizácie sa neposielajú ďalej, ani v prípade, pokiaľ takú aktualizáciu neviem rozpoznať.

Výber najlepšej cesty

BGP si vyberá optimálnu cestu pomocou algoritmu [11], ktorý prechádza zoznam kritérií zoradených podľa vopred definovanej dôležitosti, napríklad ďalší skok (*next-hop*), počet autonómnych systémov v ceste, lokálna preferencia (*Local Preference*), atď.

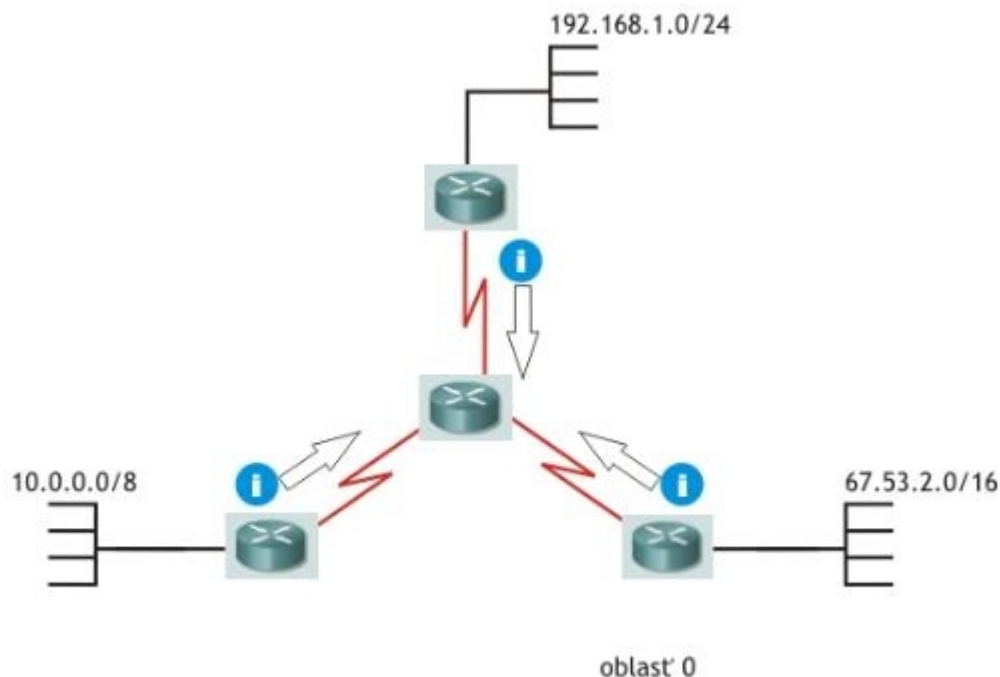
5.4 Protokoly s LS algoritmom

Základom LS protokolov je algoritmus analogický alebo príbuzný k protokolu SPF (*Shortest Path First*), ktorého autorom je Dijkstra. Základ tohto algoritmu spočíva v hľadaní najkratších ciest medzi hranami v orientovanom grafe. Špecifikácia vyžaduje, aby hrany mali len pozitívne hodnoty. Algoritmus nie je stavaný na vytvorenie celého stromu, ale na nájdenie najkratšej cesty medzi dvoma uzlami grafu, ktorý reprezentuje našu sieť.

Tento algoritmus je spravidla náročnejší na výpočet ako používajú protokoly s DV algoritmom. Tento prístup tiež vyžaduje, aby LS smerovače mali kompletný prehľad o topológii, v ktorej smerujú. Na základe toho sú schopné vnímať topológiu „z nadhľadu“ a výber najlepších ciest z topologickej tabuľky prispôbiť tak, ako by to bolo najvhodnejšie vzhľadom na polohu konkrétneho smerovača, ktorý najlepšie cesty práve kalkuluje.

Týmto štýlom spracovania sa líšia oproti DV protokolom, ktoré sa riadia technikou smerovania pomocou zvestí (*Routing by rummor*), ktoré nie sú schopné vidieť celú topológiu zo svojho pohľadu. Dalo by sa zjednodušiť povedať, že DV protokoly majú len tie informácie, ktoré im povedia ich najbližší susedia.

Aj tá najmenšia zmena topológie si vyžiada prepočet všetkých najlepších ciest v rámci celej topológie. Je to snaha, aby smerovanie bolo optimálne v rámci celej topológie. Avšak v prípade, že uvažujeme rozsiahlu topológiu s rôznymi zmenami stavov liniek, mohlo by dôjsť k nechcenému preťaženiu CPU smerovačov. Preto sa celá topológia môže rozdeliť na menšie celky, tzv. oblasti (*areas*), v ktorých sa vykonáva smerovací proces, ako to znázorňuje obr. 5.4.



Obr. 5.4: Znalosť celej oblasti pri LS protokoloch

Existuje jedna stratégia návrhu topológie siete, vďaka ktorej sa môžeme úspešne vyhnúť zložitým výpočtom SPF algoritmu. Hovorí, že oblasť (*area*) by nemala byť navrhovaná ako príliš rozsiahla, pretože SPF algoritmus sa kalkuluje vždy pre príslušnú oblasť. Nenastanú intenzívne zmeny v sieti, ktoré sú náročné pre SPF prepočet.

Očistené aktualizácie sú založené na základe zmeny siete, ktoré až pri zmene nejakej časti topológie sietí túto informáciu rozistribuujujú. Vo svete LS protokolov napriek očisteným aktualizáciám existuje akýsi mechanizmus kontroly týchto aktualizácií, tzv. Link-State obnova (*Link-State refresh*). Táto obnova distribuuje po sieti topologickú databázu vo veľmi dlhých intervaloch (štandardne 15 alebo 30 min.), ktoré nemajú efekt na zmenu využiteľnej šírky pásma liniek. Tento mechanizmus zaisťuje aj pri krátkodobom rozšírení nepresných informácií následnú správnu konvergenciu v rámci oblasti.

V Link-State architektúre je dôležité pamätať si svojich susedov, celú topológiu a najlepšie cesty, preto je založená na existencii troch tabuliek:

- **Tabuľka susedov** (*Neighbor table*): udržiava si všetkých susedov, ktorí majú nadviazanú reláciu so smerovačom
- **Topologická tabuľka** (*Topology table*): je smerovacia mapa obsahujúca každú jednu sieť v danej oblasti
- **Smerovacia tabuľka** (*Routing table*): obsahuje výber najlepších ciest z topologickej tabuľky

5.4.1 Protokol OSPF

OSPF (*Open Shortest Path First*) je protokol založený na LS algoritme. Ide o otvorený štandard, ktorý bol vytvorený IETF skupinou.

Oblasť

Aby sa však zamedzilo enormnému posielaniu LS paketov, smerovače sa zaraďujú do oblastí (*area*). Jeden smerovač môže patriť do viacerých oblastí, ale linka patrí práve do jednej oblasti. Oblasť je logické zoskupenie smerovačov a liniek. Oblasť sa správa čiastočne autonómne a mnohé správy sú vymieňané len v rámci jednej oblasti. Znižuje sa tak počet SPF (*Shortest Path First*) kalkulácií vo fáze výpočtu. Detaily zmien v topológii sa prejavujú len v rámci oblasti a zastavujú sa na hranici oblasti (*area boundary*). Rozšírenie informácie je limitované. Menší počet smerovačov taktiež znižuje počet údajov v smerovacej a topologickej tabuľke. Všetky smerovače v tej istej oblasti majú rovnakú topologickú tabuľku (*topology table*) a musia sa pripojiť do oblasti 0 (*area 0*), spravidla priamo. Sú ale možnosti, ako možno pripojiť oblasť aj nepriamo, cez inú oblasť, pomocou tunelovania cez túto oblasť.

Smerovače, ktoré sprostredkovávajú smerovanie medzi jednotlivými oblasťami sa nazývajú ABR (*Area Border Routers*) smerovače [12]. ABR smerovače sú smerovače, ktoré majú rozhrania vo viacerých oblastiach.

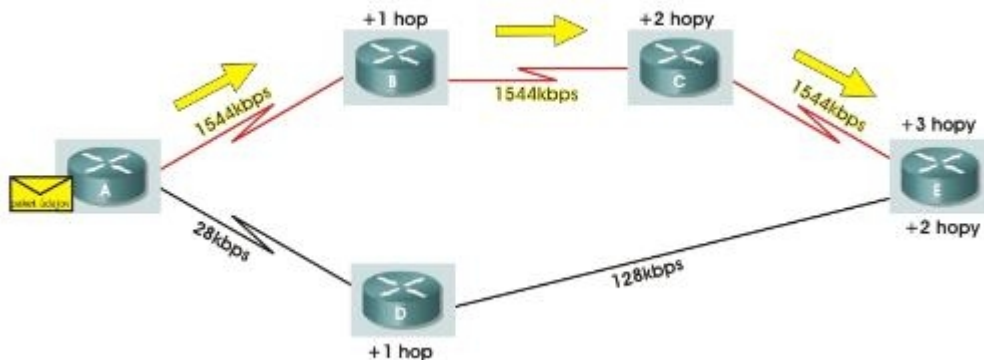
Interné OSPF smerovače sú také, ktorých rozhrania operujú len v rámci jednej oblasti, chrbticové (*backbone*) OSPF smerovače sú také, ktorých minimálne jedno rozhranie je pripojené do oblasti 0 (*area 0*).

Smerovače, ktoré sú vnímané z pohľadu topológie siete ako smerovače brány do iného autonómneho systému, sa v OSPF sieťach nazývajú ASBR (*Autonomous System Boundary Routers*).

Iba smerovače typu ABR a ASBR sú schopné sumarizovať cesty. Smerovače ABR sú výstupným uzlom zo svojej oblasti a smerovače ASBR sú výstupným uzlom pre celý autonómny systém, resp. doménu. Práve tieto smerovače by sa mali podieľať na sumarizácii.

Metrika

Výsledok výberu najlepšej cesty je v OSPF označovaný ako cena (*cost*) cesty. Ako vstupný parameter do tejto metriky bola zvolená šírka pásma. Tento parameter patrí do skupiny statických parametrov, je pevne zapísaný v konfigurácii smerovača. Obr. 5.5 ukazuje, že výber najlepšej cesty sa uskutoční na základe informácie o šírke pásma linky.



Obr. 5.5: Výber najlepšej cesty pomocou OSPF

Vzorec na výpočet ceny OSPF je zapísaný v 5.1.

$$cena = \frac{10^8}{\text{šírka pásma}} \quad (5.1)$$

Cena je výsledok nepriamej úrovne, teda čím menšia šírka pásma, tým väčšia cena. Metrika sa nepohybuje v desatinných číslach, z toho vyplýva, že maximálna šírka pásma, ktorá je rozoznateľná týmto vzorcom je 100 Mbps. Za touto hranicou majú všetky linky hodnotu 1. Tento nedostatok sa dá riešiť tým, že OSPF umožňuje pevne zapísať do konfigurácie novú hodnotu násobku, ktorý sa použije na prepočet ceny (zmena konštanty 10^8).

Vytvorenie relácie

OSPF relácie sa formujú iba v rámci rovnakej oblasti, nemožno formovať reláciu medzi rôznymi oblasťami systému. Najprv si smerovač určí vlastné identifikačné číslo, pod ktorým bude v sieti vystupovať. Spravidla ide o najvyššiu IP adresu pri štarte OSPF procesu na zariadení, pričom logické rozhrania sú vždy uprednostňované nad fyzickými rozhraniami. Identifikačné číslo môže byť aj pevne zapísané, čo je preferovaná voľba.

Následne sa začnú posielat *hello* správy (*Hello message*) na rozhraniach zahrnutých do OSPF procesu. Na to, aby bolo možné pokračovať, v prijatej *hello* správe je potrebné zaistiť, aby boli následovné atribúty rovnaké: *hello* a *router dead* časovače, typ siete, priame fyzické spojenie, číslo oblasti, typ hesla a heslo pre autentifikáciu, ak je zadané. Ďalej je to napríklad príznak listovej oblasti.

Smerovače si začnú vymieňať obsah svojich link-state databáz pomocou DBD (*DataBase Description packet*) paketov. Po tejto výmene sú smerovače synchronizované a spustí sa SPF algoritmus, ktorý vygeneruje príslušné cesty do smerovacej tabuľky.

Typy sietí v OSPF

OSPF rozoznáva niekoľko typov režimov. Podľa [RFC2328] sú to *point-to-multipoint* a *nonbroadcast* (NBMA). Cisco prišlo s ďalšími režimami: *point-to-multipoint nonbroadcast*, *broadcast* a *point-to-point*.

Smerovače si vytvárajú susedstvo s priamo spojenými smerovačmi. Na zdieľaných médiach, režimy *broadcast* a *nonbroadcast*, by sa pri väčšom počte muselo vytvoriť až príliš veľa susedstiev, čo by zaťažovalo zariadenia. Pri počte 10 smerovačov, by bolo vytvorených na každom smerovači 9 susedstiev a spolu by ich bolo 45 na médiu. Preto sa zaviedla voľba povereného smerovača (*Designated Router - DR*) a záložného DR (*Backup Designated Router - BDR*). Len DR a BDR smerovače požadujú od všetkých smerovačov vytvorenie susedstva. Ostatné smerovače (DROTHER) sa spoja len s DR a BDR. Väčšina typov OSPF paketov je posielaných na skupinových adresách. IP adresa 224.0.0.6 pre pakety smerujúce na DR a BDR a 224.0.0.5 pre všetky smerovače. V prípade výpadku DR rolu preberá BDR smerovač a začne sa voľba BDR. Voľba je nepreemptívna a ovplyvnená nastavenou prioritou. Smerovače s najvyššou prioritou preberajú rolu DR a BDR. Mali by to byť najviac výkonné smerovače na médiu.

LSA správy

OSPF rozlišuje niekoľko LSA (*Link-State Advertisement*) typov správ. Podľa toho, ktoré správy bude smerovač posilať a generovať sa odvíja od typu smerovača. Smerovače vo vnútri oblasti sa nazývajú interné a generujú typy 1 a 2. Sú to záplavové správy o všetkých cestách, ktoré pozná smerovač. Rozšírenie je v rámci oblasti po hranicu oblasti. Hraničné smerovače navyše generujú správy typu 3 – interná sumarizácia. Typ 4 je generovaný hraničnými smerovačmi, len keď je v sieti prítomný smerovač hraničiaci s iným autonómnym systémom a posila externú sumarizáciu. ASBR generuje správu typu 1 a ABR ju rozposielajú do siete ako typ 4. Typ 5 reprezentuje externé cesty do iného autonómneho systému, generované ASBR a posielané každým ABR. Z ďalších typov je zaujímavý ešte typ 7, využívaný pri technike NSSA, pozri ďalej.

Typy oblastí

Dôvod, prečo je potrebné deliť OSPF sieť do oblastí je efektívnosť. Hneď ako toto rozdelenie urobíme, smerovače v rámci jednej oblasti budú spravovať menšiu topologickú tabuľku.

Existuje niekoľko typov oblastí, ktoré sú predurčené k špecifickým podmienkam fungovania v závislosti od topológie siete.

Chrbticová oblasť (*backbone area*), alebo *area 0*, je tranzitná oblasť, do ktorej sa pripájajú ostatné, regulárne oblasti. Ak chceme, aby si dve oblasti medzi sebou vymieňali informácie, je potrebné ich pripojiť pomocou tejto chrbticovej oblasti.

Listová oblasť (*stub area*) je oblasť, do ktorej neprúdia aktualizácie s externými cestami okrem všeobecnej brány. Cieľom týchto oblastí je filtrovať smerovacie informácie vnútri oblastí, pričom filtrovanie je vykonávané blokovaním LSA typu 5 na vstupe do oblasti. Jednotlivé smerovače dostávajú zhrnuté informácie LSA typu 3. Smerovač ABR tejto listovej oblasti má stále všetky potrebné informácie, ale do vnútra listovej oblasti sa distribuuje iba všeobecná brána. Tieto listové oblasti nemôžu byť použité na tunelovanie pre nepriame dosiahnutie chrbticovej oblasti inou oblasťou.

Celkom listová oblasť (*totally stubby area*), v ktorej sú zakázané externé cesty blokovaním všetkých ciest z ostatných oblastí LSA typu 5, sú blokované aj sumarizované cesty LSA typom 3 a 4. Cieľom tohto typu oblasti je docieľiť, aby všetky smerovače poznali len naučenú všeobecnú cestu (*default route*), čo je výhodné pre úsporu šírky pásma a náročnosti smerovania na procesor smerovača. Túto požiadavku zaistí smerovač ABR vhodným filtrovaním LSA paketov. Jediná cesta ako dostať pakety z tejto oblasti je pomocou všeobecnej cesty. Opäť platí, že smerovač ABR si

ponecháva všetky informácie a LSA blokuje pre smerovače vnútri takejto oblasti. Ide o proprietárne rozšírenie.

Nie tak listová oblasť (*not so stubby area*) je oblasť, ktorá predpokladá napojenie do ešte ďalšej smerovacej oblasti [RFC3101]. Informácie z tejto ďalšej oblasti je treba oznámiť ďalším OSPF smerovačom aj mimo tejto listovej oblasti. Preto bol vymyslený LSA typ 7, ktorý pošle interný smerovač tejto listovej oblasti svojmu ABR smerovaču, ktorý tento typ LSA prijme a pošle ďalej do oblasti 0 ako LSA typ 5, ktorý býva zvyčajne označený ako externá cesta smerovaný cez ASBR. Výsledkom je, že informácie o smerovaní nadobudnuté v rámci celkom listovej oblasti sa šíria ďalej do siete. Stále pritom platí, že smerovače v takejto nie tak listovej oblasti majú odblokované LSA správy 3, 4 a 5.

Všeobecne platí, že oblasti označené ako listové a celkom listové by mali mať iba jeden ústiaci bod vo svojej oblasti. Je možné mať viacero ústiacich bodov, ale treba si uvedomiť, že potom máme viacero ABR smerovačov v týchto oblastiach, ktoré spôsobia potrebu existencie viacerých všeobecných ciest z týchto oblastí, medzi ktorými je treba vykonať vyrovnávanie záťaže. Toto vyrovnávanie záťaže je principiálne veľmi nešťastné riešenie, pretože smerovače vnútri týchto oblastí nemajú predstavu, čo sa nachádza za týmito oblasťami.

Externé cesty

V OSPF rozoznávame dva typy externých ciest, ktoré slúžia na prenos ciest z rôznych autonómnych systémov.

Cesty označované ako typ E1 (*External type 1*) sú cesty, ktorých cena (*cost*) sa zvyšuje tým, ako putujú cez autonómny systém. Je to výhodné pre systémy, ktoré majú viacero ústiacich bodov z autonómneho systému. Touto technikou je možné rozlíšiť, ktorá cesta je lepšia na toto použitie.

Cesty označované ako typ E2 (*External type 2*) sú cesty, ktorých cena (*cost*) zostáva prechodom cez autonómny systém. Je to implicitné riešenie nastavené v konfigurácii a je výhodné pre autonómne systémy s jedným výstupným bodom.

5.4.2 Protokol IS-IS

IS-IS (*Intermediate System to Intermediate System*) bol pôvodne navrhnutý ako smerovací protokol pre OSI (*Open Systems Interconnection*), čiže na svojom počiatku, keď bol vyvíjaný armádou, nebol vôbec uspokojený na chod v TCP/IP sieťach, ale pre chod v OSI. Preto bolo nutné protokol prepracovať, jeho jadro zovšeobecniť a integrovať do neho TCP/IP model. Takéto rozsiahle prepracovanie si žiadalo nové značenie, preto sa pred IS-IS operujúci v TCP/IP sieťach pridalo slovo *Integrated*, teda *Integrated IS-IS*. IS-IS je vhodný pre veľmi rozsiahle siete, keďže v jednej oblasti môže byť až vyše tisíc smerovačov. Preto sa stal obľúbeným protokolom pre poskytovateľov služby.

IS-IS je vo všeobecnosti považovaný za robustnejší s väčšou mierou individuálnej konfigurácie čít (*features*). V porovnaní s protokolom OSPF, má viac možností prispôsobenia. Avšak niektoré jeho hodnoty nie sú implicitne nastavené na optimálne, ale po ich nastavení sa protokol stáva vo všeobecnosti viac efektívny ako OSPF.

Prechod na IPv6 pri tomto protokole sa odhaduje ako nenáročný. Je to spôsobené tým, že keď sa uskutočnilo prepracovanie IS-IS na prechod do TCP/IP sietí, stalo sa jadro systému natoľko univerzálne, že prechod z IPv4 na IPv6 nebude náročný. Je to rozdiel od ostatných protokolov operujúcich v TCP/IP.

Napriek tomu, že Integrated IS-IS operuje v TCP/IP, stále pracuje s OSI adresami. OSI adresy vychádzajú zo služieb CLNS (*Connectionless Network Services*) a sú alternatívnou k IP adresám.

Smerovacie domény

Formálne, protokol IS-IS rozoznáva štyri základné druhy spojení, v ktorých nadväzuje relácie:

- **Úroveň 0** (Level 0): spojenie medzi koncovým systémom ES-IS (*End System to Intermediate System*)
- **Úroveň 1** (Level 1): spojenie medzi smerovačmi v rámci oblasti
- **Úroveň 2** (Level 2): spojenie medzi smerovačmi medzi oblasťami
- **Úroveň 3** (Level 3): spojenie medzi autonómnymi systémami

V ďalšom uvažujeme len úrovne 1 a 2, keďže ide o základné úrovne používané pri smerovaní.

Úrovne smerovania

IS-IS je LS protokol, ktorý používa SPF (*Shortest Path First*) algoritmus pre výber najkratšej cesty. Na zisťovanie prítomnosti svojich susedov používa *hello* správy.

IS-IS preferuje rozdeliť autonómny systém do menších skupín – oblastí. Každá z týchto oblastí má vlastnú topológiu. Existuje preto niekoľko úrovní, v ktorých smerovače pôsobia:

- **Úroveň 1** (Level 1): Smerovače si spravujú topológiu iba vo svojej vlastnej oblasti
- **Úroveň 2** (Level 2): Smerovače pôsobiace v chrbtici siete (*backbone*), vedia iba o cestách, ktoré sú pridané do chrbtice siete
- **Úroveň 1/2** (Level 1/2): Spravujú databázy oboch úrovní

Databázy úrovne 1 a úrovne 2 sú kompletne separované. Smerovače úrovne 1 nadväzujú relácie vo svojej oblasti a medzi oblasťami sa nadväzujú vzťahy úrovne 2. Možno povedať, že smerovače úrovne 2 sú niečo ako preddefinovaná brána pre smerovače typu úrovne 1.

Metrika

Podľa RFC 1142 IS-IS rozoznáva štyri podtypy výpočtu najlepšej cesty [RFC1142]. Každý IS (*Intermediate System*) má povolené počítat metriku minimálne jedným podtypom. Pritom každý IS by mal byť schopný počítat metriku cez predvolenú metriku. Avšak rôzne IS môžu počítat najlepšiu cestu rôzne.

- **Predvolená metrika** (*default metric*): každej linke by mala byť priradená celistvá hodnota predstavujúca ľubovoľnú veličinu, napríklad šírku pásma
- **Metrika oneskorenia** (*delay metric*): je to voliteľná časť metriky predstavujúca oneskorenie medzi linkami
- **Metrika nákladov** (*expense metric*): pracujúca na základe vrstvy monetárnych cien. Je to voliteľná časť výpočtu najlepšej cesty. Podľa RFC konvencie je hodnota 1 používaná na indikáciu voľného spojenia. Vyššia hodnota indikuje väčšiu monetárnu cenu.
- **Chybová metrika** (*error metrics*): pri každej linke sú detegované chyby. Je to nepovinná časť výpočtu najlepšej cesty.

Proces kombinácií týchto metrík je cez jednoduché sčítanie hodnôt jednotlivých podtypov. Preto je dôležité tieto hodnoty voliť primerane vzhľadom k ostatným zadaným hodnotám.

Cisco smerovače implementujú len predvolenú metriku. Na informáciu o výsledku výpočtu metriky je vyhradený 1 bajt. Ôsmy bit je rezervovaný. Siedmy bit rozlišuje medzi internou a externou klasifikáciou. Interná metrika je metrika v rámci IS-IS domény, externá metrika je mimo IS-IS domény alebo naučená od iného sieťového protokolu. Na hodnotu výpočtu metriky ostáva 6 bitov. Preto na hodnotu IS-IS metriky zostáva iba interval $\langle 0; 63 \rangle$. Celková suma jednotlivých metrík na linkách nemôže presiahnuť hodnotu 1023. Takáto metrika sa nazýva úzka (*narrow*). Rozšírená metrika, nazývaná ako široká, je 24 bitová pre linku a 32 bitová pre súčet cien na linkách a teda podporuje lepšiu granularitu pri určovaní výslednej ceny.

Pri implementácii metriky v TCP/IP sieťach na smerovačoch firmy Cisco výpočet metriky neobsahuje informáciu o šírke pásma. Metrika obsahuje iba implicitnú hodnotu priradenú každej linke o hodnote 10, čo spôsobuje, že pri nezmenení týchto hodnôt sa tento protokol správa ako protokol RIP, teda vzdialenosti porovnáva počtom smerovačov od štartu k cieľovej sieti.

Vytváranie susedstiev

IS-IS pri vytváraní susedstiev taktiež využíva *hello* pakety. Na zdieľanom médiu existuje DIS (*Designated Intermediate System*), čo je to isté ako DR pri OSPF. Záložný DIS sa nevolí a voľba je preemptívna. Navyše susedstvá si vytvorí vždy každý s každým. *Hello* pakety sú posielané na DIS smerovače každých 3,3 sekundy, ostatným každých 10 sekúnd. Na nie zdieľanom médiu je to tiež každých 10 sekúnd. Interval *router dead* je nastavený na trojnásobok *hello* času. Celá databáza sa odosiela raz za 15 minút na kontrolu konzistentnosti.

Stavba smerovacej tabuľky

V mnohých implementáciách smerovače hľadajú najlepšiu cestu pre IS-IS založenú na cene (*cost*) získanej pomocou SPF algoritmu. Túto činnosť vykonávajú smerovače úrovne 1 aj úrovne 2 kompletne samostatne. Potom sú všetky najlepšie cesty vsunuté do smerovacej tabuľky OSI. Z tejto OSI smerovacej tabuľky sa za pomoci PRC (*Partial Route Calculation*) algoritmu vyberajú cesty pre smerovacie tabuľky IP. Všetky najlepšie cesty získané PRC algoritmom sa finálne zahrnú do IP smerovacej tabuľky.

6 Konvergencia na sieťovej vrstve

Optimalizácia na sieťovej vrstve je rozsiahla a to predovšetkým u smerovacích protokolov, ktoré sú priamo stavané na doladovanie. Smerovacie protokoly obsahujú mnohé časové parametre, ktorých implicitné hodnoty sú už pre dnešnú dobu príliš konzervatívne a preto je všeobecne odporúčané si ich zmeniť. Popíšeme podstatné časové a nečasové parametre, ktoré je odporúčané zmeniť a pozrieme sa bližšie aj na ich negatívne dôsledky. Zameriame sa na LS protokoly, keďže ako jediné sú schopné kooperovať s MPLS TE. Popíšeme aj zásady dobrého návrhu siete na logickej úrovni, čo ešte viac zlepši časy konvergencie. Treba si uvedomiť, že po optimalizácii budeme mať rýchlejšie konvergujúcu sieť, čo zvýši dostupnosť uzlov. Vyššia dostupnosť znamená vyššiu kvalitu služieb a vyššiu spokojnosť zákazníka.

Táto kapitola je priamo prebraná a upravená z [7].

Dosiahnutie konvergencie na sieťovej vrstve je komplexná záležitosť. Na sieťovej vrstve zabezpečujeme smerovanie pomocou smerovacích protokolov. Konvergencia sa stáva dominantnou témou a reflektuje schopnosť siete, ako rýchlo sa dokáže prispôbiť zmenám v sieti.

Konvergencia je proces, kedy sa všetky uzly dohadujú na optimálnych cestách pre smerovanie paketov a vykonajú následnú aktualizáciu ich smerovacích tabuliek. Ak nastane akákoľvek zmena na akomkoľvek smerovači, alebo aj inom zariadení, sieť už nie je v skonvergovanom stave a je nutná opätovná konvergencia. Sieť môže mať porušený tento stav z nasledujúcich príčin:

- pridanie, odobratie alebo výpadok linky
- pridanie, odobratie alebo výpadok zariadenia
- pridanie, odobratie alebo výpadok segmentu siete
- zmena protokolu alebo niektorých jeho parametrov
- administratívny zásah, vypnutie linky alebo zariadenia z dôvodu údržby, ľudský faktor

Na sieťovej vrstve a pri smerovaní sa stretávame s viacerými variantmi pojmu konvergencia. Len pripomenieme, že sieť považujeme za skonvergovanú, keď všetky uzly dosiahli konsenzus pri nových hodnotách a všetky príslušné údaje v smerovacích tabuľkách sú aktualizované [13].

Konvergencia protokolu (*protocol convergence*) – vyjadruje proces, v ktorom si uzol aktualizuje svoju smerovaciu tabuľku a príslušné zmeny oznámi svojim susedom. Od momentu, kedy sa proces ukončil, je protokol skonvergovaný.

Konvergencia cesty (*route convergence*) – vyjadruje proces, v ktorom si uzol aktualizuje svoju smerovaciu tabuľku a dáta sú presmerované na novú cestu. Od momentu, kedy sa proces ukončil, je cesta skonvergovaná.

Konvergencia siete (*network convergence*) – vyjadruje proces, v ktorom si všetky uzly aktualizujú svoje smerovacie tabuľky. Od momentu, kedy sa proces ukončil, je sieť skonvergovaná.

Konvergencia siete je buď [7]:

- suma konvergencií všetkých protokolov (KP): $\sum_{i=1}^n KP_i$
- maximum z množiny konvergencií protokolov: $\max\{KP_1, KP_2 \dots KP_n\}$
- kombinácia predošlých dvoch prípadov

Ako vidieť z definícií, celkový pojem konvergencia, sa dá na sieťovej vrstve rozdeliť na menšie fragmenty. Čas medzi porušením skonvergovaného stavu a opätovným nadobudnutím tohto stavu označujeme ako čas konvergenzie. Konvergencia nikdy nie je okamžitá. Otázka je, aký dlhý čas je potrebný na dosiahnutie tohto stavu [13].

6.1 Konvergencia v sieti

Počas konvergenzie je pravdepodobné, že nastane stratovosť paketov na určitú dobu. V najhoršom prípade to je 100 %-ná stratovosť počas celej periódy. Výhodou rýchlej konvergenzie v sieti je krátka perióda zvýšenej stratovosti paketov.

Konvergencia v sieti prebieha v niekoľkých krokoch [14]:

- Detekcia
- Rozšírenie informácie
- Výpočet
- Aktualizácia smerovacej tabuľky

Časový rozdiel medzi zmenou a detekciou tejto zmeny v sieti je fázou detekcie. Jej trvanie môže byť veľmi krátke, desiatky milisekúnd, pokiaľ hovoríme o reštarte protokolu, vypnutí linky a podobne. V najhoršom prípade sú to desiatky sekúnd až minúty, ak ide napríklad o výpadok zariadenia a protokol to zistí až po niekoľkonásobnom neprijatí *hello* paketov (všeobecne interval *router dead*). Práve tu je obrovský priestor na optimalizáciu. Rýchlosť detekcie ovplyvňuje aj typ interfejsu. Napríklad POS (*Packet Over SONET*) interfejs dokáže zistiť výpadok za niekoľko milisekúnd a nie je ani nutné optimalizovať interval *router dead*. Optimalizácia pozitívne vplyva na reakciu pri sledovaní objektov vo VRRP protokole, popísané nižšie. Rýchla detekcia znamená kratší čas nedostupnosti koncových zariadení a aj rýchlejšiu konvergenciu.

Rozšírenie informácie o zmene je závislé od väčšieho množstva parametrov. Najmarkantnejším parametrom je počet smerovačov, ktoré túto správu musia prijať. Správny výber topológie a predovšetkým hierarchický návrh siete so sumarizáciou smerovacích ciest na smerovačoch zabezpečí nutnosť rozšírenia správy len po určitú oblasť. Pokiaľ by sa nevyužila sumarizácia, správa by musela byť rozšírená možno až do celej siete. Pri niektorých protokoloch musia túto správu prijať všetky smerovače v rámci celej oblasti. Rozšírenie informácie ovplyvňuje rýchlosť liniek a zahltenie. Predchádzaniu zahltenia sa podrobnejšie venujeme v kapitole 7 o QoS. Ďalším parametrom je použitý protokol a jeho optimalizácia. Celkový čas rozšírenia informácie zjednodušene môžeme vyjadriť vzťahom:

$$(\text{priemerné oneskorenie na každom uzle}) * (\text{diameter siete}) \quad (6.1)$$

Rýchlosť vo fáze výpočtu závisí od konkrétneho protokolu a implementovaného algoritmu a od aktuálnej záťaže na smerovači. Problém nastáva, ak má byť vykonaných viac výpočtov v krátkej dobe. Napríklad protokol OSPF obmedzuje implicitne výpočet na max. jedenkrát za 5 sekúnd, pričom aj prvý výpočet je pozdržaný. Parameter sa dá optimalizovať. Takéto obmedzenia v protokoloch mali v minulosti zabrániť zbytočnej záťaži pre smerovače pri viacnásobných prepočtoch, avšak v dnešnej dobe je odporúčané takéto hodnoty znížiť alebo anulovať.

Posledným krokom k dosiahnutiu skonvergovaného stavu je aktualizácia smerovacej tabuľky, prípadne všetkých smerovacích tabuliek, ak ich smerovač využíva viac. Čas je závislý od rýchlosti a vyťaženia zariadenia. Takisto od veľkosti smerovacej tabuľky a od počtu ciest, ktoré je nutné aktualizovať. Veľkosť smerovacej tabuľky vieme ovplyvniť využívaním techník ako

sumarizácia, listová oblasť, alebo statické smerovanie. Snažíme sa udržiavať čo najmenej ciest v smerovacej tabuľke.

Konvergencia na sieťovej vrstve prebieha v niekoľkých krokoch, pričom v každej fáze je možná optimalizácia, aby sme dosiahli kratší čas celkovej konverencie. Výsledný čas je však silne závislý od nižších vrstiev. Na sieťovej vrstve máme k dispozícii viac nástrojov na optimalizáciu. Výber vhodného smerovacieho protokolu, jeho časová optimalizácia, používanie rozšírených možností techník protokolu, sumarizácia IP adries a iné prostriedky na zmenšenie smerovacej tabuľky, QoS a mnohé iné pozitívne vplyvajú na čas konverencie [15].

6.2 Časové parametre smerovacích protokolov

V tejto kapitole sa bližšie pozrieme na časové parametre definované v smerovacích protokoloch, ktoré je možné a vhodné optimalizovať. Zmena časových parametrov nasleduje až po vyčerpaní všetkých ostatných techník definovaných smerovacím protokolom. Presný návod, aké hodnoty nastaviť neexistuje, ale sú všeobecné odporúčania. Najväčšie množstvo časových parametrov je práve v smerovacích protokoloch.

Implicitné časové hodnoty definované v protokoloch boli v dobe svojho vzniku udávané ako optimálne. Keďže protokoly „starnú“ a rýchlosti, či už prenosového pásma, alebo CPU, sa zvyšujú, môžeme jednotlivé časové parametre optimalizovať a nastaviť nižšie hodnoty. Dokonca niektoré parametre je možné anulovať, keďže svoj význam postupne strácajú. V minulosti veľké východzie hodnoty časov slúžili ako ochrana proti nadmernej záťaži zariadení. Vzhľadom na výkon dnešných zariadení sú tieto ochranné parametre takmer zbytočné. Chybovosť a stratovosť na linkách je omnoho menšia, čiže aj spoľahlivosť siete sa zvýšila. Vyššia spoľahlivosť znamená menej takzvaných falošných pozitívnych správ. Pri oscilácii linky (*flapping link*) [16], kedy linka rýchlo mení svoj stav z aktívneho na neaktívny, by mohla byť vyvolaná opätovná konvergencia. Avšak oscilácia linky väčšinou znamená hardvérovú poruchu, ale môže to byť aj problém s nastavením taktovacieho kmitočtu (*clock rate*), prípadne duplicitná IP adresa na linke. Takéto oscilácie môžu byť najväčším strašiakom po optimalizácii, keďže ich detekcia a následné spracovanie bude veľmi rýchle. Obranou je sumarizácia IP adries a tlmenie interfejsu (*dampening*).

Zmena časových parametrov je asi najjednoduchšou technikou optimalizácie a aj veľmi účinnou. Každý časový parameter môže byť zmenený. Rozdielne je už len rozpätie. Skracovanie časov v dnešnej dobe umožnili niekoľkonásobne rýchlejšie linky a zariadenia. Skracovanie časov síce umožní rýchlejšiu detekciu, rýchlejšie rozšírenie správy a výpočet, ale zvýši záťaž na linke a smerovači.

Smerovacie protokoly používajú viacero parametrov, ktoré sú nastaviteľné. Tieto parametre chránia samotný protokol a zariadenie pred zahltením v časoch nestability siete. Dá sa povedať, že sú istou prevenciou. Väčšina parametrov je vo svojej východzej konfigurácii nastavená značne konzervatívne. Je to preto, lebo v minulosti mala prednosť stabilná sieť pred rýchlou konvergenciou. Pri vzniku protokolov boli iné podmienky, aké sú dnes. Zariadenia a linky sú rýchlejšie, média v sieťach LAN operujú väčšinou v plne duplexnom režime a média sú väčšinou nesúperiace. Chybovosť pri prenose je zanedbateľná a celkovo stabilita siete sa zvýšila. Navyše novšie implementácie majú priamo v nastavovaní časových parametrov zabudovanú obranu proti nestabilite vo forme tlmenia. To všetko umožňuje zmenu časových parametrov na omnoho agresívnejšie hodnoty. Naším cieľom bude dosiahnutie času konverencie pokiaľ možno pod jednu až dve sekundy na sieťovej vrstve v smerovacích protokoloch. Podmienkou je možnosť v IOS daného zariadenia nastaviť takúto hodnotu. Pokiaľ zariadenie nemá možnosť nastavenia času pod

jednu sekundu, je nevyhnutné nainštalovať novšiu verziu. Je možné, že všetky parametre, ktoré popíšeme sa nenachádzajú v danej verzii IOS_u, alebo nie sú prístupné pre danú platformu. V takom prípade sa daný parameter nebude optimalizovať. Syntax, alebo dokonca názov parametra sa môže líšiť medzi jednotlivými verziami a platformami.

Názvy techník budú vychádzať z IOS c3660-jk9o3s-mz.124-12.bin. Platforma je určená pre stredne veľké siete, verzia je 12.4 a ponúka mnohé rozšírenia a nadštandardné možnosti. Vybrali sme si tento IOS práve preto, aby sme na jednej verzii demonštrovali všetky možnosti optimalizácie. Budeme popisovať iba LS protokoly OSPF a IS-IS, keďže iba tie vedia pracovať s MPLS TE, čo je popísané nižšie.

6.2.1 Protokol OSPF

V prvom rade sa pozrieme na fázu výpočtu. Vo fáze výpočtu sa využíva SPF algoritmus. Problémom je, že fáza výpočtu sa začne až po 5 sekundách od prijatia požiadavky na prepočet! Takéto zamedzenie malo predchádzať viacerým prepočtom v krátkej dobe v prípade nestability na sieti. Neskoršie implementácie však umožňujú dynamické prispôsobovanie čakania na prepočet. Konfiguruje sa pozdržanie prvého prepočtu, pozdržanie medzi prvým a druhým prepočtom a maximálny čas pozdržania. Pozdržanie medzi druhým a tretím prepočtom a každým ďalším prepočtom v krátkej dobe, je pozdržaný o dvojnásobok predošlej hodnoty, až po maximálny čas pozdržania. Parameter je veľmi výhodné optimalizovať.

Ďalším vylepšením je ISPF (*Incremental SPF*). Po aktivácii tejto možnosti algoritmus neprepočítava celý strom, ale len zmeny v strome. Ak by bol nutný prepočet celého stromu, použije sa klasické SPF. Časové parametre pre SPF sa vzťahujú aj na ISPF. Aktivácia zabezpečí rýchlejší prepočet a rýchlejšiu konvergenciu, pričom nemá žiaden negatívny vplyv. Pokiaľ je táto možnosť k dispozícii, rozhodne by mala byť aktivovaná [17].

Na podobnom princípe pracuje aj škrtenie LSA (*LSA throttling*). Podstatou je obmedziť posielanie LSA správ susedom. Prvá správa sa vždy posiela okamžite. Na totožnom princípe ako v predošlom prípade funguje princíp pozdržania odosielania LSA správ.

Príchod LSA správ (*LSA arrival*) zakazuje prijatie tej istej LSA správy na nejaký čas. Tá istá LSA správa má rovnaké LSA ID, LSA typ a je prijatá z rovnakého smerovača.

Implicitná hodnota posielania *hello* paketov je 30 sekúnd na WAN a na ostatných typoch liniek 10 sekúnd. Tento parameter sa zdá byť nie až taký dôležitý, ale priamo s ním súvisí interval *router dead*. Ten je nastavený na štvornásobok *hello*. Znížením času *hello* sa skráti čas vo fáze detekcie. Rýchlejšia detekcia akou je jedna sekunda nie je možná, keďže interval *router dead* je minimálne jedna sekunda. Zmeniť sa dá len periodičita posielania *hello* paketov – násobič. Čím väčší násobič, tým viac *hello* paketov je poslaných za jednu sekundu. Paradoxne, čím menší násobič, tým je štatisticky rýchlejšia konvergencia, keďže prijatie *hello* paketu nastaví časovač na 1000 ms. Pri násobiči 20 by sa časovač pohyboval v rozpätí 950 – 1000 ms. Pri násobiči 4 je časovač v rozpätí 750 – 1000 ms. Zmena tohto parametra je najviac nebezpečná, keďže permanentne vyťažuje zariadenie a linku. Príliš agresívna hodnota môže spôsobiť v časoch nestability na sieti, alebo v čase vysokej záťaže, falošné správy, ktoré by zbytočne vyvolali opätovnú konvergenciu. Po zmene tohto parametra je nutné sledovať správanie sa zariadení na sieti, či nové hodnoty akceptujú. V súbehu s ostatnými technikami, ktoré popisujeme, by sa výraznejšie problémy nemali objaviť.

Po nastavení hodnôt pod jednu sekundu je odporúčané nastaviť ešte tlmenie na danom interfejsi. Tlmenie v časoch nestability na danom interfejsi exponenciálne pozdržuje čas zmeny

stavu linky. Každá zmena stavu linky vyvoláva opätovnú konvergenciu. Tlmenie je teda ochranou interfejsu a aj ochranou siete. Jej zapnutie na interfejsi vysoko odporúčame na všetkých interfejsoch, pretože obráni sieť pred zbytočnou konvergenciou a teda udržuje úroveň stability s pred optimalizácie.

OSPF ponúka kategóriu základných parametrov (*padding*) na optimalizáciu. Všetky tieto parametre by mali byť ponechané na implicitných hodnotách a radšej využiť techniky ako sumarizácia, listová oblasť a podobne. Preto sa týmito parametrami nezaobráame. Navyše neexistuje návod, ako tieto hodnoty zmeniť, keďže sú priamo závislé od návrhu siete.

Pri OSPF odporúčame jednoznačne zmeniť parametre pri SPF a zapnúť ISPF. Výsledkom sú lepšie časy konvergencie vo fáze výpočtu. Zmena *hello* a intervalu *router dead* zlepši časy detekcie. Aby sa neznížila stabilita interfejsu, zariadenia a siete, treba nutne zapnúť tlmenie. Škrtenie LSA a príchod LSA správ urýchlia časy rozšírenia správ.

6.2.2 Protokol IS-IS

IS-IS bol vo svojom vzniku navrhnutý tak, aby ho bolo možné optimalizovať a prispôbovať.

Tak ako OSPF aj IS-IS využíva SPF algoritmus, ktorý je však pozdržaný až 5,5 sekúnd. Princíp je zhodný s OSPF.

IS-IS taktiež umožňuje aktivovať ISPF. Narozdiel od OSPF je možné pozdržať aktiváciu tejto techniky od jej nakonfigurovania [18].

IS-IS dokáže pomocou použitia techniky PRC prepočítať zmenu v strome bez použitia SPF. Táto možnosť je asi 10x rýchlejšia ako klasické SPF a využíva sa vtedy, keď mechanizmus smerovania sa nezmenil, ale smerovač predsa len nejakú zmenu zaznamenal. Napríklad zmenu prefixu.

Protokol vie pozdržať generovanie LSP (*Link State Packet*) správ podobne ako pri OSPF.

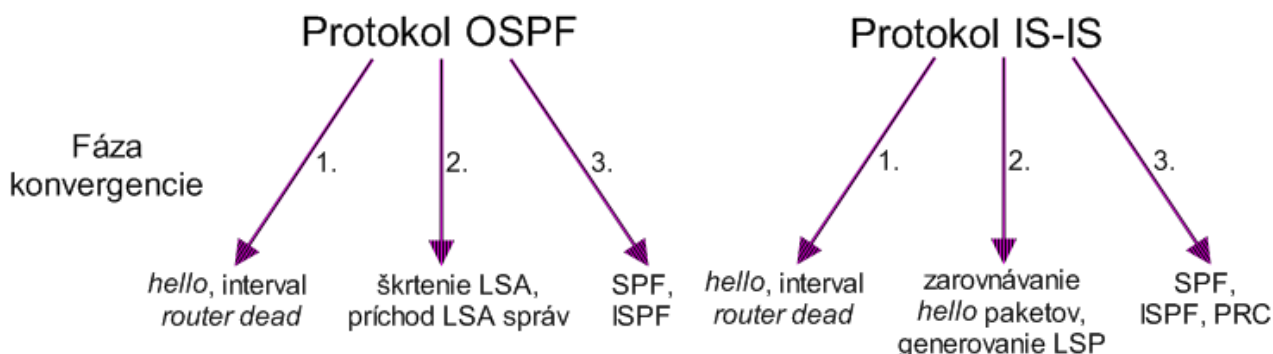
Hello pakety sa odosielajú každých 10 sekúnd a na DIS každých 3,3 sekundy. Opäť platí všetko to, čo bolo popísané pri OSPF. Rozdielna je syntax a taktiež možnosť posielat *hello* správy až každú milisekundu. Interval *router dead* je trojnásobok *hello* [19].

Možnosti zmeny SPF, PRC a zapnutie ISPF zlepšia fázu výpočtu v konvergencii. Zmena *hello* a *dead* intervalu pod jednu sekundu zabezpečí lepšie časy detekcie, no je nutné zapnúť na interfejsoch tlmenie, aby sme stabilizovali sieť v prípade problémov. Generovanie LSP správ ovplyvňuje fázu rozšírenia správy do siete.

Protokol má v niektorých implementáciách implicitne zapnuté zarovnávanie *hello* paketov na dĺžku MTU (*Maximum Transmission Unit*) - *hello padding*. Túto možnosť je vhodné vypnúť, keďže nie je potrebná a zbytočne konzumuje príliš veľkú časť šírky pásma [20].

Na obrázku 6.1 sme zhrnuli všetky optimalizačné nástroje oboch smerovacích protokolov a priradili sme ich ku konkrétnej fáze konvergencie. Na základe tejto schémy budeme v návrhu optimalizovať jednotlivé parametre a študovať posun v konvergencii.

Optimalizácia fáz smerovacích protokolov



Obr. 6.1: Optimalizácia fáz smerovacích protokolov

6.2.3 Protokol RSVP

Protokol RSVP (*Resource Reservation Protocol*) slúži na signalizáciu požiadaviek medzi uzlami v sieti. Požiadavky sú smerované cestami, ktoré boli vybrané smerovacími protokolmi, preto je vhodné začať s jeho optimalizáciou až po skončení optimalizácie smerovacích protokolov. Tento protokol môže byť taktiež optimalizovaný. Protokol RSVP sa používa v modeli *IntServ* v kvalite služieb a v MPLS TE, čo je popísané v nasledujúcich kapitolách.

Optimalizované môže byť množstvo odosielaných správ, ich veľkosť a počet vo fronte. Takisto sa môže kontrolovať signalizácia a navyše sa môže optimalizovať úvodné pozdržanie signalizácie v prípade opakovaného vysielania, čakanie počas ktorého sa akumulujú potvrdenia (občerstvenie) a iné. Pomocou aktivovanej predikcie kompresie je možné redukovať réžiu UDP alebo RTP transportného protokolu.

Takisto je možné optimalizovať *hello* protokol a *keepalive* správy v protokole RSVP. Ich podporu sme v našom sledovanom IOS_e však nenašli. Je to preto, lebo náš IOS nepodporuje funkcie rýchleho presmerovania. Interval vypršania sa určuje ako počet neprijatých *hello* paketov. Ďalšie možnosti optimalizácie sa buď nevzťahujú na IP protokol, alebo nie sú momentálne pre nás podstatné [21].

6.2.4 Protokol LDP

Protokol LDP (*Label Distribution Protocol*) signalizuje správy a stará sa o výmenu návěsti v prostredí MPLS. Bližšie sa čitateľ o tomto protokole dozvie v kapitole o MPLS.

Protokol LDP umožňuje optimalizovať svoj *hello* mechanizmus a časovač *holddown*. Ide o podobný mechanizmus ako v smerovacích protokoloch. Ďalej je možné upraviť časovač pre udržovanie relácie po výpadku. Veľmi dôležitá funkcia je IGP (*Interior Gateway Protocol*) synchronizácia, ktorá je však podporovaná len v LS smerovacích protokoloch. Táto funkcia aktivuje synchronizáciu medzi IGP (označované tiež ako IRP) a LDP a predíde tak strate paketov, ktoré by mohli byť smerované na základe IGP protokolu, ale nevedia byť prepnuté na základe návěstia. Existujú prípady, kedy IGP má vytvorené susedstvo, ale LDP susedstvo sa zrušilo a komunikácia sa stráca. Takisto táto funkcia predchádza chybám v implemetáciach alebo v konfiguráciach [22].

Keďže funkcia pozdrzuje konvergenciu IGP a eventuálne pozdrží aj konvergenciu LDP, tento príkaz sa zdá byť pre nás menej vhodný. Jeho vhodnosť prakticky verifikujeme.

Veľmi zaujímavou možnosťou je ochrana relácie v LDP protokole. Ak vytvorené susedstvo medzi dvomi smerovačmi by malo byť zrušené napríklad výpadkom linky, avšak susedia sú navzájom dosiahnuteľní alternatívnou cestou, ktorá je známa, tak sa relácia nezruší. Susedia sú dosiahnuteľní cez cieľené odosielanie *hello* paketov a navzájom o svojej prítomnosti vedia. Po obnovení linky nie je nutné nanovo vytvárať susedstvo a vymieňať si stavové informácie. Tento proces totižto spomaľuje konvergenciu, preto túto funkciu odporúčame aktivovať. Zároveň šetríme prostriedky, keďže smerovače nemusia nanovo vytvárať návestia a oznamovať ich susedovi a eventuálne do siete [23].

V tejto časti sme popísali základné a podstatné časové parametre v protokoloch, ktoré budeme využívať pri implementácii, ich princíp a fungovanie. Parametrov na optimalizáciu je omnoho viac, no nie sú relevantné alebo podstatné. Táto kapitola je nosná pri optimalizácii siete s cieľom zrýchliť konvergenciu a zvýšiť dostupnosť uzlov.

6.3 Stabilita po časovej optimalizácii

Ako sme si všimli, aj samotné nastavenia protokolov prechádzajú postupnou evolúciou. V minulosti sa mnohé parametre konfigurovali len zmenou jednej hodnoty. Avšak dnes, ako napríklad SPF a LSA príkazy, majú už komplexnejšiu syntax. Zmena pozdržania SPF kalkulácii mala len jeden parameter, ktorý sa vzťahoval na všetky SPF kalkulácie. Jeho zmena na príliš nízke hodnoty v čase nestability siete by mohla mať výrazný negatívny vplyv na zariadenie a aj sieť ako takú. Avšak dnešné implementácie majú až tri parametre pre jeden príkaz. Všeobecne to je pozdržanie prvej kalkulácie, rozdiel medzi prvou a druhou kalkuláciou a maximálne pozdržanie. Pričom každá nová požiadavka na prepočet je pozdržaná o dvojnásobok času hodnoty predošlej kalkulácie počnúc časom medzi prvou a druhou kalkuláciou. Takéto správanie sa nápadne podobá na BGP tlmenie, odkiaľ mnohé smerovacie protokoly túto vlastnosť prebrali. Práve takáto úprava nám zabezpečí nastavenie nulovej hodnoty pri pozdržaní prvého prepočtu a pomerne nízku hodnotu medzi prvým a druhým prepočtom. Už sa nemusíme obávať, že by nestabilita mala devastujúci účinok, protokol sa vysporiada s nestabilitou sám.

Ďalším dôkazom evolúcie sú aj možnosti konfigurácie posielania *hello* paketov pod jednu sekundu. Keďže takéto generovanie a spracovanie paketov je veľmi rýchle, preto by určite mala byť zapnutá možnosť tlmenia na interfejsi. Opäť to isté správanie ako pri BGP tlmení. Nestabilita interfejsu v podobe oscilujúcej linky nám nezahltí sieť falošnými správami na opätovnú konvergenciu.

7 Kvalita služieb

V tejto kapitole budeme diskutovať o potrebe kvality služieb, spôsoby, algoritmy a technológie na jej dosiahnutie.

Ako aplikácie postupne umocňujú expanziu a vývoj komunikačných sietí, stávajú sa požiadavky na klasifikáciu a kategorizáciu aplikačných tokov podľa potreby danej služby stále viac viditeľnými. Aplikácie s rôznymi potrebami a požiadavkami na sieť vyžadujú istú mieru inteligencie v sieťovej technológii, ktorá by vedela zaručiť splnenie takýchto požiadaviek s konečnými kapacitnými zdrojmi siete. Sieť musí zvládať zaručiť predvídateľnú kvalitu služieb pre takzvané business-critical aplikácie, aby naplnila svoju existenčnú podstatu. Zároveň umožniť aplikáciu služieb s vysokou pridanou hodnotou ako napr. VoIP, ktoré vyžadujú garanciu časového oneskorenia pri prenose. Tieto požiadavky nás privádzajú k pojmu kvalita služieb alebo *Quality of Service* (QoS). Kvalita služieb pre komunikačné siete predstavuje súbor nástrojov, ktorých aplikovaním je možné dosiahnuť determinizmus správania sa siete a zabezpečenie požadovanej úrovne kvality služieb v prípade vyťaženia siete.

V rámci nasledujúcich častí tohto dokumentu budú predstavené základné architektúry prístupu ku kvalite služieb v TCP/IP sieťach (*Best Effort*, *Integrated Services - IntServ* a *Differentiated Services - DiffServ*) a niektoré najdôležitejšie nástroje na ich implementáciu.

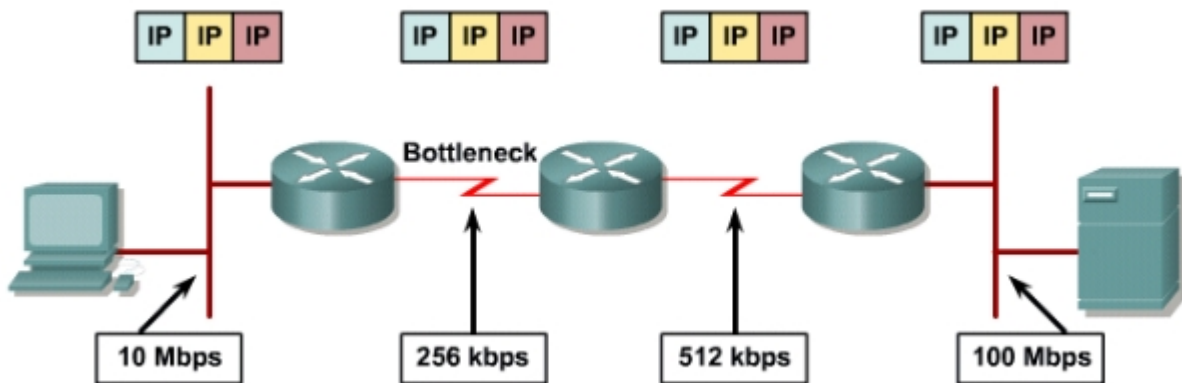
7.1 Potreba kvality služieb

Siete typu TCP/IP pri svojej prevádzke sprostredkovávajú prenosovú kapacitu pre služby, ktoré sa v sieti vyskytujú. Avšak pri prenose dátových jednotiek, rámcov, môžu nastať komplikácie vyplývajúce z fundamentálneho princípu fungovania počítačových sietí. Môže ísť o nasledovné problémy:

- Nedostatok transportnej kapacity
- Oneskorenie prenosu medzi koncovými bodmi prenosu
- Variabilné oneskorenie, takzvaný *Jitter*
- Stratú dátových jednotiek komunikácie

7.2 Nedostatok transportnej kapacity

Každá TCP/IP sieť má konečnú prenosovú kapacitu. Taktiež prenosová kapacita jednotlivých jej úsekov sa môže líšiť. Pre dátové toky prechádzajúce sieťou je však celková kapacita rovná najmenšej priepustnosti v rámci toku. Príklad možno vidieť na nasledovnom obrázku č. 7.1.

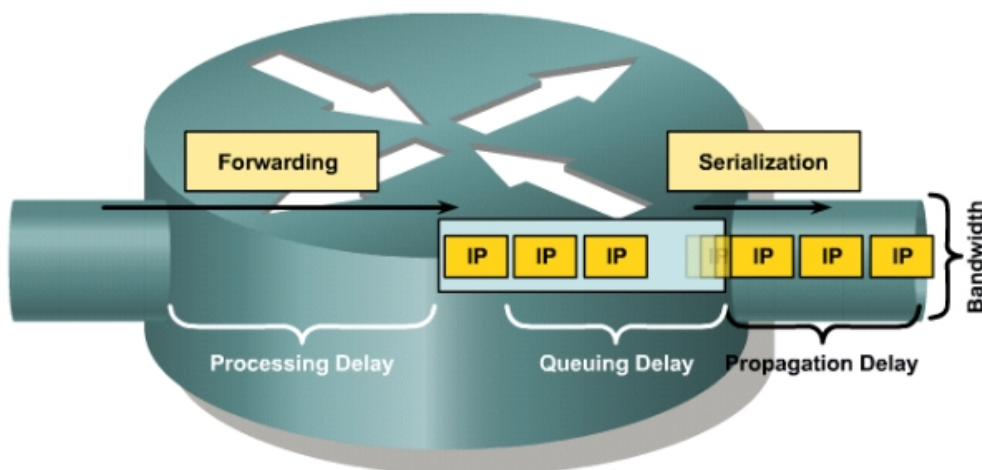


Obr. 7.1: Problém „úzkého hrdla“ alebo Bottleneck [24]

Z obrázku je zrejmé, že dátové toky medzi stanicou a serverom budú musieť zdieľať prenosovú kapacitu najmenej výkonnej prenosovej kapacity, takzvaného „úzkého hrdla“ siete alebo tiež *Bottleneck*. V prípade, ak sieť nerozpoznáva rôzne priority pre dátové toky, prenosová kapacita je jednoducho podiel prenosovej kapacity vydelený počtom tokov. Bez aplikácie nástrojov kvality služieb je jediným riešením zväčšenie kapacity siete, čo je však finančne náročné, ale v neposlednom rade tiež nemôže zaručiť kvalitu služieb v prípade neprirodzenej prevádzky v sieti.

7.3 Oneskorenie

Existujú služby v TCP/IP sieťach, pre ktoré je oneskorenie kľúčovým parametrom pri prevádzke. Znamená to, že pri prekročení istej hodnoty oneskorenia sa stáva ich fungovanie menej kvalitným, až neakceptovateľným. Opäť, príkladom môže byť VoIP, alebo aj videohovory alebo hry pre viacerých hráčov. Oneskorenie vzniká na viacerých miestach v prevádzke sieťových komponentov, obr. 7.2.



Obr. 7.2: Rôzne miesta oneskorenia v sieťových komponentoch [24]

Základná kategorizácia zdrojov oneskorenia je teda nasledovná:

- **Oneskorenie spracovania** (*Processing Delay*) je oneskorenie spracovania dátového toku alebo rámca logikou sieťových zariadení
- **Oneskorenie fronty** (*Queuing Delay*) je oneskorenie, kedy rámec dátového toku je už spracovaný logikou sieťového zariadenia a bol zaradený do frontu, v ktorom čaká na príležitosť, aby bol vyslaný po dátovej linke
- **Oneskorenie serializácie** (*Serialization delay*) je oneskorenie prenosu logických dát rámca na samotné fyzické médium dátovej linky
- **Oneskorenie propagácie** (*Propagation delay*) je oneskorenie vznikajúce na základe času potrebného na prešírenie prenosového signálu na transportnom médiu

7.3.1 Variabilné oneskorenia

Oneskorenie v sieti sme si už rozobrali, na druhej strane je potrebné si uvedomiť, že aj keď sú služby schopné prispôbiť sa istej miere konštantného oneskorenia, variácie v tomto oneskorení môžu predstavovať prílišnú prekážku v ich správnom fungovaní.

7.4 Straty v toku dát

Straty sú fundamentálne najhorším nepriateľom pre kvalitu tej danej služby. V prípade strát je možné, že aplikácie bude žiadať preposlanie stratených dát, čo znamená viacnásobne zaťaženie kapacity siete tými istými dátami. Taktiež aplikácie ako VoIP potrebujú svoje dáta v predpokladanom čase oneskorenia a pokiaľ ich v danom čase nedostanú, preposielanie stráca na význame a hlasová komunikácia trpí radikálne zhoršenou kvalitou.

7.5 Modely prístupu ku kvalite služieb

Kvalita služieb je rozsiahla tematika s pomerne početnou skupinou nástrojov na dosiahnutie vytýčených cieľov pre kvalitu služieb. Z pohľadu nasadzovania týchto nástrojov do siete je potrebné zvoliť adekvátny architektonický prístup. V súčasnosti sa abstraktné prístupy k architektúre nasadzovania kvality služieb delia na tri rôzne druhy, známe ako *Best Effort*, *Integrated Services* a *Differentiated Services*.

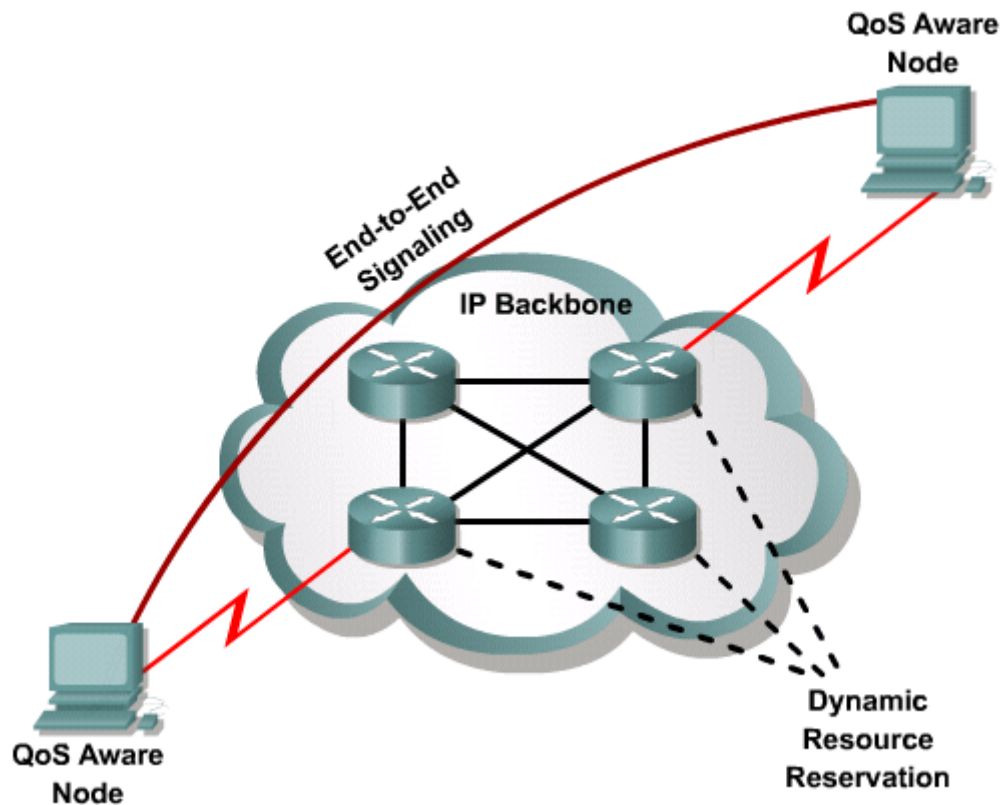
7.5.1 Model *Best Effort*

Model *Best Effort*, čo v preklade znamená „Najlepšia snaha“ predstavuje referenčnú predstavu o sieti, v ktorej sa nenachádzajú komplexné nástroje na riadenie kvality služieb. Znamená to, že v takejto sieti nie sú dátové toky jednotlivých aplikácií diferencované z pohľadu siete a všetky dosahujú rovnakú dostupnosť k sieťovým prostriedkom. Sieť s týmto prístupom sú postačujúce pre služby špecifickej potreby na prenosové kapacity. Príkladom takýchto služieb sú napríklad klasické dátové toky ako sú toky aplikácii pre email, web a textovú komunikáciu. Spoločným znakom týchto služieb je, že v prípade nedostatkov prenosových kapacít v sieti, sú tieto služby schopné, aj keď sú limitované, istým spôsobom splniť svoju funkciu. Na druhej strane sieť typu *Best Effort* nie sú vhodné pre služby citlivé na prenosové kapacity ako je napríklad VoIP. V prípade zahltenia siete,

napr. služba VoIP, vie deterministicky splniť svoju funkciu, nakoľko pokiaľ nie sú pre jej dátový tok splnené minimálne prenosové požiadavky, stráca svoj účel a jej používatelia nie sú schopní sa dorozumieť.

7.5.2 Model *IntServ*

Táto architektúra sa zakladá na možnosti dynamickej indikácie požiadaviek pre prenosové kapacity koncových staníc. Znamená to, že koncová stanica nahlási najbližšiemu sieťovému zariadeniu svoje požiadavky. Toto sieťové zariadenie požiadavku akceptuje (ak môže) a nahlási tieto požiadavky ďalšiemu sieťovému zariadeniu na ceste do cieľa toku dát. Každé zariadenie na ceste musí požiadavku akceptovať, aby bolo možné zabezpečiť požadovanú kvalitu. Až keď všetky zariadenia súhlasia s požiadavkami, vtedy sa pre daný tok vyčlení prenosová kapacita a zdrojová stanica môže začať komunikovať. Rezerváciu zdrojov zabezpečujú špecifické protokoly (ako RSVP) a daný protokol musí podporovať každé zariadenie v sieti vrátane koncových staníc. Na nasledujúcom obrázku je ukážka tohto princípu.

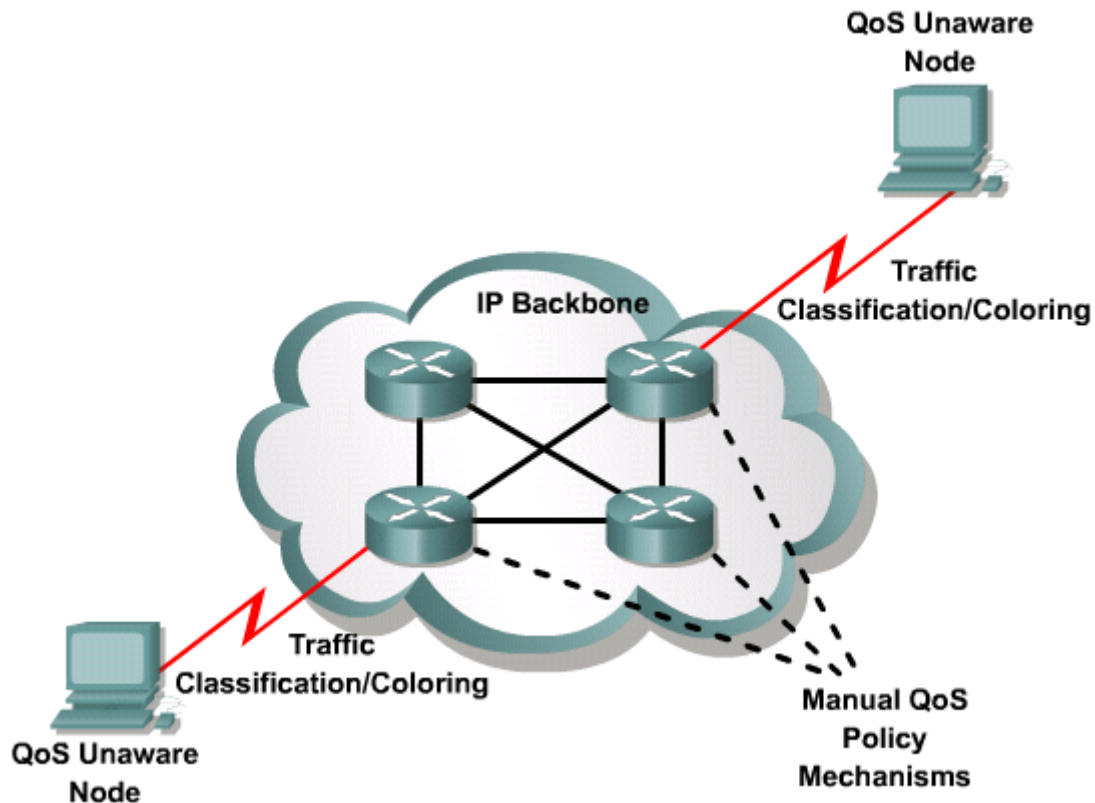


Obr. 7.3: Model architektúry *IntServ* [24]

7.5.3 Model *DiffServ*

Narozdiel od *Integrated services*, v tomto modeli nie sú požiadavky vôbec hlásené medzi zariadeniami. V *Differentiated Services* sú nástroje zabezpečenia kvality služieb integrované do každého sieťového zariadenia samostatne. Koncové stanice môžu byť do tohto procesu tiež zaradené, ale ide o sporadické riešenie. Výhodou tohto riešenia je jeho veľká škálovateľnosť,

flexibilita a kontrola nad kvalitou služieb. Tá je riadená iba sieťovými zariadeniami a nie koncovými stanicami, ktoré sú často pokladané za nedôveryhodné. Samotné dátové toky sú klasifikované do kategórií s priradenou kvalitou služieb podľa potreby danej služby v danej kategórii. Taktiež sa v tejto architektúre intenzívne používa značkovanie samotných dátových jednotiek (IPv4/IPv6 ToS, 802.1Q CoS, MPLS EXP), ktoré umožňuje, aby klasifikácia prebiehala podľa komplexných pravidiel iba na okrajových zariadeniach siete a jadro siete nestrácalo čas analýzou tokov. Na nasledujúcom obrázku je zobrazený princíp tejto architektúry.



Obr. 7.4: Model architektúry DiffServ [24]

7.6 Nástroje *Differentiated Services*

V rámci implementácie modelu *Differentiated Services* v tejto prípadovej štúdie je potrebné poznať nástroje, ktorých celková aplikácia vytvorí tento model. Tieto nástroje budú na nasledujúcich stránkach analyzované z pohľadu na umožnenie optimalizácie konvergencie a optimalizácie siete s podporou VoIP a iných multimediálnych dátových tokov v sieti.

7.6.1 Značenie paketov

Aby sa nemuselo na každom smerovači pre prichádzajúci paket zisťovať, do ktorého radu má byť zaradený, existuje technika značkovania paketov. Značkovanie prebehne len na jednom smerovači a ostatné smerovače v sieti budú akceptovať a veriť označeniu. Týmto ušetríme čas pri šírení informácie. V IP protokole na to slúži položka IP precedense, respektíve TOS (*Type Of*

Service). Ide o osembitové pole, ktoré označuje prioritu. V iných protokoloch existujú iné podobné polia s podobnou funkciou. Všeobecne platí, čím vyššia hodnota tohto poľa, tým je paket dôležitejší. Jednotlivé modely QoS využívajú toto pole rôzne a rôzne označujú pakety. Definovaných je len 8 základných tried. Trieda 7 a 6, teda dve najvyššie triedy, sú rezervované. Ostatné triedy je možné použiť na označovanie bežnej dátovej komunikácie. Trieda 7 je rezervovaná pre signalizáciu a trieda 6 pre smerovacie protokoly. Čiže pri výbere akéhokoľvek značenia, pakety smerovacích protokolov označujeme triedou 6.

7.6.2 Manažment radov

QoS je možné implementovať s rôznymi algoritmami na manažment radov. Významné algoritmy popíšeme z pohľadu vhodnosti pre smerovacie protokoly.

Najtriviálnejším algoritmom je algoritmus **FIFO** (*First In First Out*), ktorý prichádzajúce pakety v rovnakom poradí posiela do výstupných vyrovnávacích pamäti. Takýto prístup je nevhodný, nakoľko v skutočnosti žiaden manažment nepredstavuje.

PQ algoritmus (*Priority Queuing*) definuje štyri rady pre pakety. Každý z prichádzajúcich paketov je označený a zaradený do jedného z týchto radov. Až keď je rad s najvyššou prioritou vyprázdnený, obsluži sa nasledujúci rad. Algoritmus je vyhovujúci, pokiaľ sa v najviac prioritnom rade nachádzajú len pakety smerovacích protokolov. Celkovo však algoritmus môže spôsobiť starváciu pre ostatné dáta.

Nasledovníkom PQ algoritmu sa stal **CQ** algoritmus (*Custom Queuing*). Definuje 16 radov a každá trieda paketov má pridelené isté množstvo priestoru v rade. Rady sú obsluhované cyklicky, čím sa vytratila schopnosť striktnej priority a tak je tento algoritmus pre konvergenciu menej výhodný.

Algoritmus **WFQ** (*Weighted Fair Queuing*) zabezpečuje spravodlivý podiel šírky pásma pre všetky toky. Prioritnejšie toky dostávajú väčší podiel zo šírky pásma. Problém, ktorý vznikol pri CQ algoritme však naďalej pretrváva a ani tento algoritmus z pohľadu konvergence nie je vhodný, nakoľko neumožňuje definovanie vlastných priorít pre rôzne triedy dátových tokov.

CBWFQ (*Class-Based WFQ*) algoritmus zaisťuje spravodlivosť a garantuje prenosové pásmo pre jednotlivé rady. Tento algoritmus umožňuje definovať triedy dátových tokov a prioritizovať medzi nimi, avšak ani tento algoritmus nevie pracovať s paketmi citlivými na oneskorenie a preto ani tento algoritmus neodporúčame v rámci tejto prípadovej štúdie.

Pridaná podpora uprednostňovania paketov do CBWFQ vytvorila algoritmus **LLQ** (*Low-Latency Queuing*), taktiež známy ako PQ-CBWFQ. Ide o kombináciu PQ algoritmu, ktorý zabezpečí uprednostnenie vybraných dátových tokov a CBWFQ algoritmus, vhodný pre ostatné dátové toky. Podobne ako pri PQ je nutné si dávať pozor, koľko a ktoré dátové toky majú byť striktno uprednostnené. Voľba tohto algoritmu je výhodná nielen z pohľadu konvergence v sieti, ale aj pre ostatné dátové toky a v rámci tejto prípadovej štúdie bude tento algoritmus implementovaný v sieti pre podporu multimediálnych tokov pre ich potrebu minimalizovať oneskorenie.

7.6.3 Techniky predchádzania zahltenia

Na jednotlivých zariadeniach môže dochádzať k zahlteniu. O zahltení hovoríme, ak tok prijímaných dát je rýchlejší ako tok odosielaných dát. Dôsledkom je zaplnenie vyrovnávacej pamäte a jej následné pretečenie, čo znamená stratovosť paketov. QoS obsahuje viacero techník ako včasne predchádzať zahlteniu na zariadení. Pre dáta, zabezpečujúce konvergenciu, je vhodné, aby neboli

vyhadzované a teda aby nedochádzalo k zahlteniu v ich rade. Tomu sa dá predísť rezerváciou dostatočnej šírky pásma pre daný rad. I napriek tomu je nutné uvážiť správny nástroj na predchádzanie zahltenia, keďže pri úplnom zahltení vstupnej vyrovnávacej pamäte môže dôjsť k pretečeniu a teda neprijímaniu paketov dôležitých pre rýchlu konvergenciu alebo vyvolanie falošných výpadkov v multimediálnom toku.

Vyhadzovanie paketov, až po úplnom zahltení, sa nazýva *tail drop*. Najlepšie nástroje sú algoritmy, ktoré predchádzajú zahlteniu, čiže ešte predtým, než sa vyrovnávacia pamäť úplne naplní, preventívne túto pamäť vyprázdňujú. Najlepšími algoritmami sú algoritmy včasnej náhodnej detekcie (*Random Early Detection* – RED). Algoritmus RED náhodne vyhadzuje pakety z radu od určitej nakonfigurovanej hodnoty zaplnenia vyrovnávacej pamäte. Tu je dôležité, aby pakety smerovacích protokolov neboli vyhadzované. Toto technika RED nedokáže zabezpečiť, respektíve iba v najnovšej implementácii, avšak rozlišuje len dva typy radov. Vylepšením je technika WRED (*Weighted RED*). Táto technika uvažuje už aj o precedencii a teda pri vyšších triedach môžeme znížiť agresivitu vyhadzovania. Výber modelu QoS nie je pre rýchlu konvergenciu pri smerovacích protokoloch podstatný, podstatná je garancia prenosového pásma a urýchlené preposlanie vybraných paketov. Pri použití DiffServ modelu sa treba sústrediť na správny algoritmus manažmentu radov – LLQ a aplikovať WRED na včasné predchádzanie zahltenia. Smerovacie protokoly musia byť správne označované a smerovače musia s týmto označením správne narábať. QoS redukuje oneskorenie vo vyrovnávacej pamäti, ale pridáva oneskorenie pri spracovaní paketu v uzle, keďže je potrebné pakety analyzovať a značiť, čiže prepisovať. I napriek tomu je priam nevyhnutné aplikovať QoS.

8 Redundancia na sieťovej vrstve

Redundancia na tretej vrstve RM OSI zaisťuje dostupnosť IP adries aj po výpadku fyzického uzla siete. Je to spôsobené virtualizáciou IP adresy. Pokiaľ uvažujeme o virtualizácii IP adresy, v dôsledku enkapsulácie paketu sa virtualizuje aj MAC adresa.

Existuje viacero protokolov, ktoré túto redundanciu podporujú. Možno využiť komerčné riešenie od firmy Cisco nazvané HSRP (*Hot Standby Router Protocol*) alebo GLBP (*Gateway Load Balancing Protocol*). Alternatívne k tomu môžeme využiť riešenie VRRP (*Virtual Router Redundancy Protocol*). K výhodám použitia VRRP nepochybne patrí to, že je to otvorený štandard. Pomocou VRRP môžeme virtualizovať aj polia UNIXových serverov tak dobre, ako predvolenú bránu.

8.1 Protokol VRRP

Pôvodne bolo VRRP navrhnuté na eliminovanie chyby jedného bodu (*single point of failure*). Často sa používa pre potreby eliminácie výpadku predvolenej brány. VRRP špecifikuje protokol, ktorý dynamicky priradí zodpovednosť na jeden z VRRP smerovačov na jednom sieťovom segmente [RFC2338].

8.1.1 Terminológia v protokole VRRP

Virtuálny Smerovač (*Virtual Router*) je zariadenie vytvorené na základe operácie jedného alebo viacerých zariadení s VRRP implementáciou [25]. Je to virtuálne zariadenie, ktoré reprezentuje funkcionality celého klastra sieťových zariadení a je zodpovedné za sieťovú prevádzku, ktorá prináleží každému zariadeniu na serverovom klastri.

Jednoznačný identifikátor virtuálneho smerovača (*Virtual Router ID*) je jedinečne priradené číslo na rozlíšenie fyzických VRRP zariadení. VRID musí byť jedinečné pre každý sieťový segment.

Virtuálna MAC adresa (*Virtual MAC address*) je automaticky priradená MAC adresa na VRRP rozhranie založená na klasickej MAC prefixe pre VRRP pakety a VRIP čísla. Prvých 5 oktetov je 00:00:5E:00:01 a posledný oktet je konfigurovaný ako VRID [26].

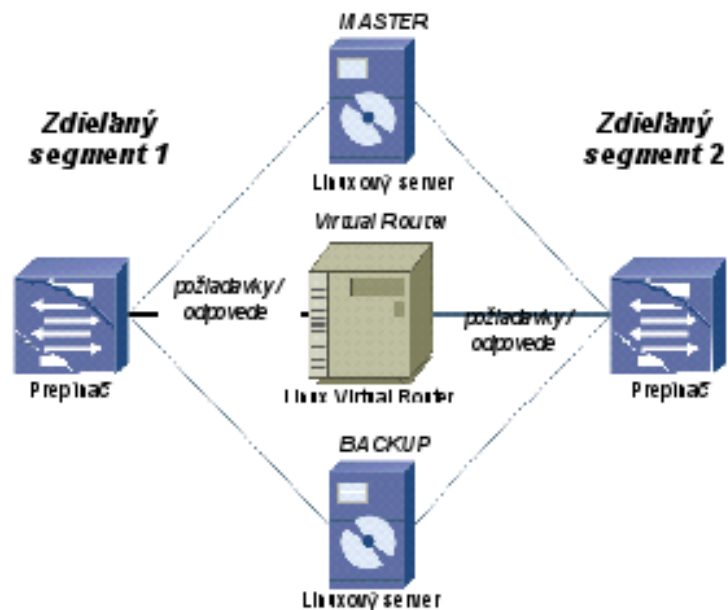
IP adresa virtuálneho smerovača (*Virtual Router IP*) tzv. VIP je IP adresa priradená k virtuálnemu smerovaču.

Master rola je zariadenie, ktoré práve reprezentuje virtuálny smerovač a vykonáva funkcionality virtuálneho smerovača [25]. Tiež prijíma a vysiela VRRP pakety na zistenie štatútu zariadení nachádzajúcich sa v roľe *Backup*.

Backup rola je zariadenie, ktoré čaká na vykonávanie funkcionality virtuálneho smerovača v prípade, že zariadenie v roľe *Master* zlyhá [25]. Zariadenie posiela a prijíma VRRP pakety na zistenie, či je zariadenie v roľe *Master* aktívne.

Priorita je číslo z intervalu 1 až 254, ktoré môže pomôcť k voľbe zariadenia v roľe *Master*. Vyššie číslo priority udáva preferenciu pri výbere *Master* zariadenia.

8.1.2 Príklad použitia



Obr. 8.1: Modelový prípad pre redundanciu

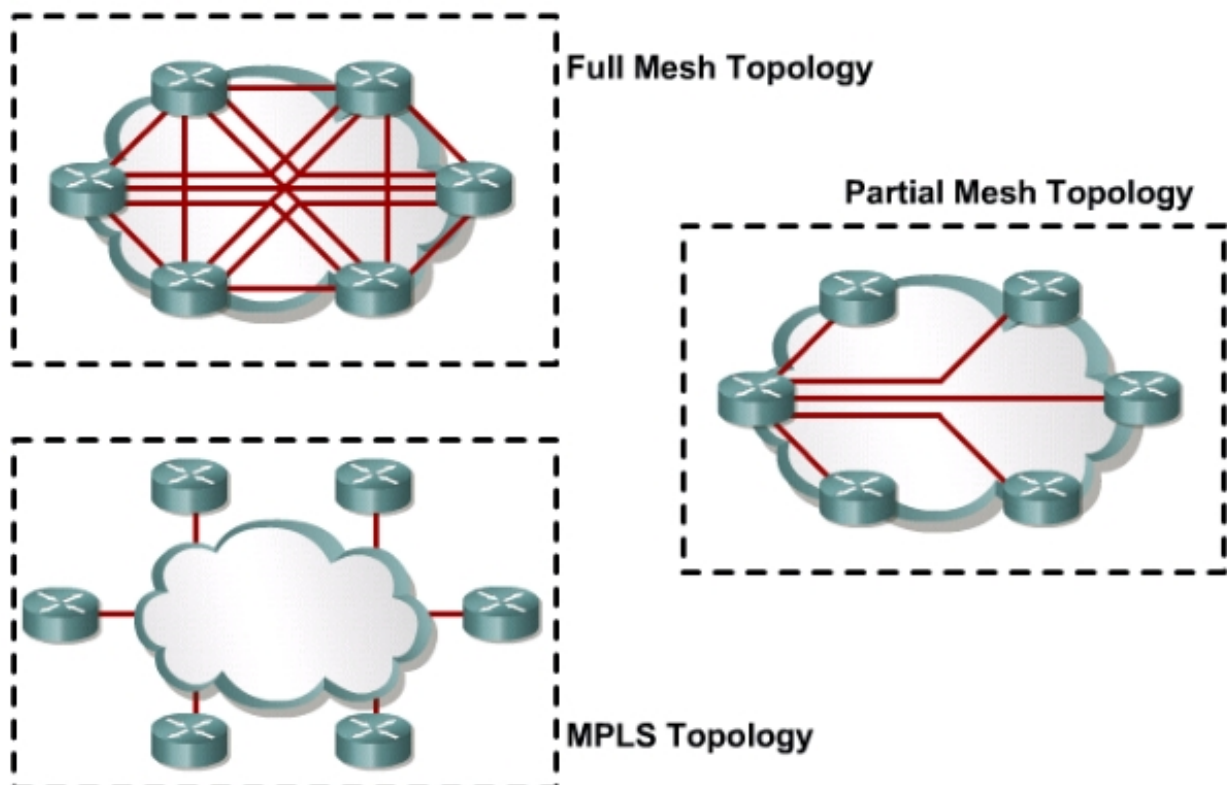
Obrázok popisuje hore uvedený princíp fungovania VRRP. Pre každý zdieľaný segment topológie sa zvolí zariadenie v *Master* role reprezentujúce virtuálny smerovač. V prípade výpadku zariadenia v role *Master* sa použije záložné zariadenie v role *Backup*. Tento výpadok je zdetegovaný neprijatím viacerých *hello* paketov na zdieľanom segmente topológie. V prípade obnovenia zariadenia v role *Master*, toto zariadenie znovu prevzme prevádzku, lebo má väčšiu prioritu.

9 Technológia MPLS

Multiprotocol Label Switching (MPLS) je technológia prepínania na základe návěstí. Spája inteligenciu smerovania na tretej vrstve RM OSI modelu s rýchlosťou prepínania na druhej vrstve RM OSI modelu. Je to teda ako keby prepínanie na tretej vrstve. Tento prístup umožňuje skombinovať viacero výhod v rámci architektúry sietí poskytovateľov sieťovej konektivity. Technológia MPLS podporuje dve implementácie a to režim prepínania rámcov (*Frame Mode MPLS*) a režim prepínania buniek (*Cell Mode MPLS*). Nakoľko režim prepínania buniek je špecifické pre ATM siete, o ktorých v tomto dokumente neuvažujeme, zameriame sa len na režim prepínania rámcov. Z pohľadu zadania tímového projektu je technológia MPLS dôležitá z toho dôvodu, že poskytuje možnosti QoS medzi koncovými bodmi komunikácie. Taktiež v tomto dokumente budú spomenuté základy implementácií MPLS sietí na podpore škálovateľných VPN sietí [RFC3031].

9.1 Úvod do MPLS

Ako je známe, na fundamentálnu schopnosť smerovať dáta optimálne, je potrebné mať konektivitu z každého bodu siete do každého ďalšieho. Keď si však predstavíme momentálne existujúce spôsoby stavania sietí typu WAN, veľa implementácií toto nespĺňa. Tento fundamentálny prínos sa snaží poskytnúť práve MPLS technológia. Pohľad na tento rozdiel medzi úplnou *mesh* (*full-mesh*) topológiou, čiastočnou *mesh* (*partial-mesh*) topológiou a MPLS topológiou je znázornený na obr. 9.1.



Obr. 9.1: MPLS maskujúci skutočnú fyzickú topológiu [27]

Ako na predošlom obrázku vidno, topológie, akou je čiastočný *mesh*, poskytujú neoptimálne smerovanie len medzi niektorými bodmi. Optimálne smerovanie poskytujú úplný *mesh* a MPLS topológia. Je potrebné si uvedomiť, že MPLS túto výhodu poskytuje tak, že zákazník si od svojho poskytovateľa nekupuje konektivitu medzi dvoma bodmi, ale kupuje si prístup k prístupovým bodom do spoločnej MPLS siete. To znamená, že zákazník sa nestará o samotné spojenia medzi svojimi bodmi, kedy mu čiastočná *mesh* topológia vychádzala lacnejšie na počet liniek. Ak si od poskytovateľa zákazník kúpi iba konektivitu pre svoje prístupové body a tiež mu poskytovateľ zabezpečí vymieňanie si informácií smerovacích protokolov medzi zákazníkovými hraničnými smerovačmi, zákazník získa z jeho pohľadu úplné *mesh* spojenie svojich bodov, avšak za cenu čiastočných *mesh* spojení. Takéto riešenie predstavuje obrovské výhody pre zákazníka pri smerovaní medzi svojimi vzdialenými sieťami a tiež pre poskytovateľa, nakoľko odpadá potreba manuálne definovať smerovanie cez prenosovú MPLS sieť.

9.2 MPLS návestia

Tento termín označuje MPLS 32-bitovú hlavičku v technológii MPLS. Táto hlavička sa pridáva za poslednú hlavičku druhej vrstvy RM OSI modelu a pred prvú hlavičku tretej vrstvy RM OSI modelu, obr. 9.2.



Obr. 9.2: Hlavička MPLS návestia [27]

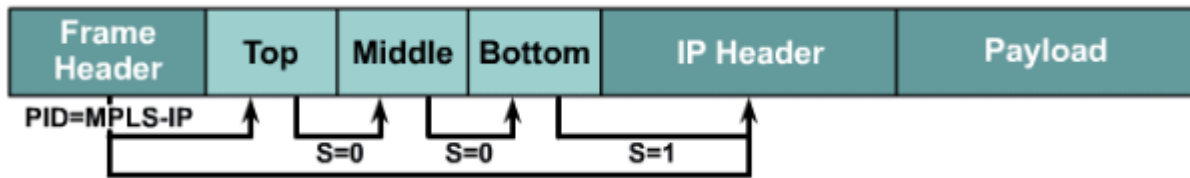
Ako z obrázka 9.2 vidno, hlavička MPLS sa skladá z viacerých polí [29]. Ich funkcia je nasledovná:

- **Label** – je 20-bitová číselná hodnota používaná na smerovanie v MPLS sieti
- **EXP** – je trojbitové pole, ktoré zatiaľ nie je definované štandardom a je rezervované pre budúce rozšírenia. V rámci implementácie na zariadeniach spoločnosti Cisco Systems a iných, je toto pole využívané ako CoS (*Class of Service*) pole pre hodnoty označenia priority v rámci zabezpečovania kvality služieb
- **S** – je jednobitová informácia indikujúca posledné MPLS návestia v rámci zásobníka návěstí. Bližšie vysvetlenie tohto dôležitého poľa bude poskytnuté v dokumente nižšie
- **TTL** – toto 8-bitové pole zabezpečuje pre MPLS sieť rovnakú funkčnosť ako TTL pole v IP hlavičke, čiže určuje živosť. TTL je v MPLS návěstí potrebné, nakoľko smerovanie sa uskutočňuje na základe MPLS návěstia a IP hlavička (a jej TTL) je ignorovaná. Každý smerovač dekrementuje túto hodnotu o jedničku, pokiaľ nie je nastavené ignorovanie. Ak sa dosiahne hodnota menšia ako jeden, celý rámec je zahodený

9.3 Vnorené MPLS návestia

MPLS technológia umožňuje veľmi elegantný systém, ako pridať dodatočne význam pre návestia v MPLS hlavičke, nakoľko umožňuje do jedného rámca pridať viacero MPLS návěstí.

Smerovače potom používajú len prvú MPLS hlavičku a ďalšie MPLS hlavičky môžu predstavovať inú ako smerovaciú informáciu. Príkladom môže byť informácia o príslušnosti dát do určitého logického tunela alebo kategórie pre kvalitu služieb. Tento spôsob umožňuje implementáciu ďalších veľmi atraktívnych riešení ako sú privátne siete v MPLS - MPLS VPN alebo MPLS TE (*Traffic Engineering*). Ukážku ako MPLS umožňuje existenciu viacerých hlavičiek vidno na obr. 9.3.



Obr. 9.3: Reťazenie viacerých MPLS hlavičiek [27]

Ako vidieť na obr. 9.3, medzi rámec a paket boli vložené tri MPLS hlavičky. Informáciu, že rámec obsahuje MPLS hlavičku nesie identifikátor v poliach hlavičiek na druhej vrstve RM OSI modelu. Zároveň tento identifikátor tiež indikuje, že za MPLS hlavičkou (alebo hlavičkami) nasleduje IP hlavička. Systém teraz môže začať spracovávať MPLS hlavičku. Pri jej spracovaní narazí na pole "S", kde nájde bit nastavený na 0. Ten indikuje, že za aktuálne spracovávanou hlavičkou je ďalšia vnorená MPLS hlavička. Toto vnorenie môže byť samozrejme viacnásobne a pokračuje až do poslednej MPLS hlavičky, ktorej pole S má hodnotu 1. V rámci tejto práce budú spracované možnosti MPLS TE. V rámci MPLS TE sa využívajú vnorené hlavičky MPLS, ktorých dodatočná informácia sa však už nepoužíva na smerovanie. Techniky MPLS TE budú v tomto dokumente podrobnejšie rozobrané v ďalších kapitolách. Ďalším možným využitím, ktoré sa v MPLS často využíva je MPLS VPN (*Virtual Private Network*). V MPLS VPN indikuje druhá vnorená hlavička lokálne jedinečné číslo tunela medzi dvoma destináciami, ktorý tvorí privátne spojenie medzi dvoma sieťami [28].

Výmenu a šírenie návěstí zabezpečuje protokol LDP (*Label Distribution Protocol*). Ide o otvorený štandard. Známe sú aj iné proprietárne protokoly, ktoré sa už však v dnešnej dobe nepoužívajú.

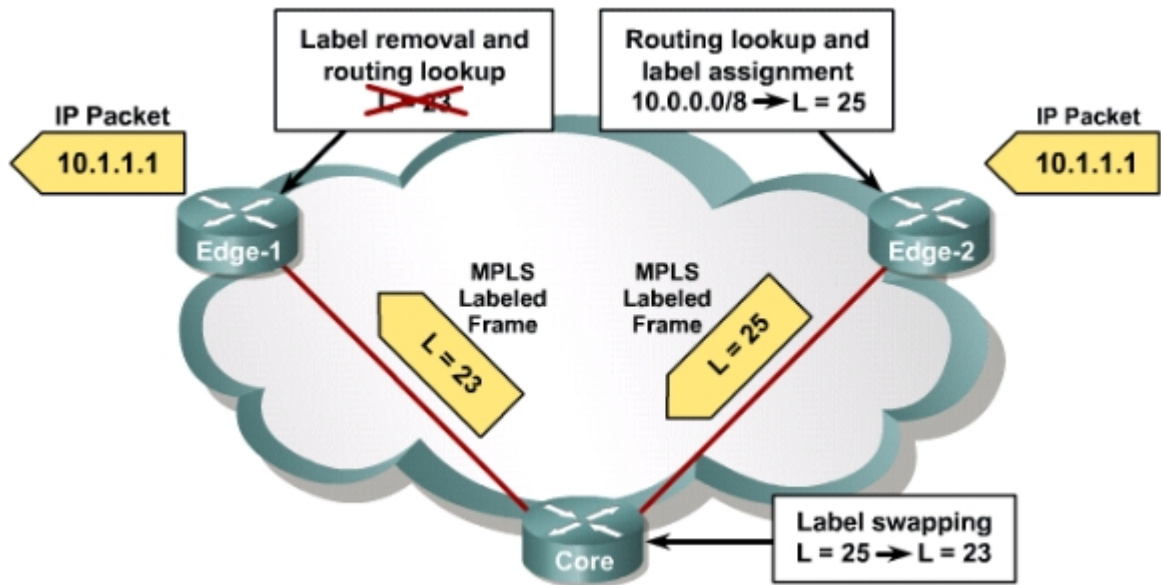
9.4 MPLS smerovanie

Technológia MPLS, ako už bolo spomenuté, vychádza zo základu, že prepínanie rámcov je rýchlejšie ako smerovanie, nakoľko je možné ho implementovať hardvérovo. Znamená to, že prepínanie, ktoré sa uskutočňuje na základe statickej MAC adresy, je efektívnejšie ako smerovanie na základe IP adresy. Preto technológia MPLS skĺbila rýchlosť a jednoduchosť prepínania z druhej vrstvy RM OSI modelu s možnosťou logickej adresácie ciest z tretej vrstvy RM OSI modelu.

Všetky tieto vlastnosti predstavuje MPLS návěstie (*Label*), ktoré je často kategorizované ako hlavička dva a pól-tej vrstvy RM OSI. MPLS sieť prepína výhradne na základe tejto MPLS hlavičky [29]. Znovu je potrebné pripomenúť, že MPLS je prenosová sieť poskytovateľa a smerovanie je založené na MPLS hlavičke pridávanej do každého rámca. Avšak ide o sieť poskytovateľa služieb a platnosť MPLS hlavičiek končí, keď rámec príde na okraj MPLS siete. Tu je MPLS hlavička odstránená a smerovanie do siete zákazníka pokračuje pôvodnou technológiou. Preto MPLS doména nedosahuje koncové stanice, ani sieť zákazníka a nie sú potrebné z jeho strany žiadne technologické úpravy ohľadom MPLS.

MPLS sieť rozdeľuje svoje smerovače na dve hlavné kategórie a to hraničný (*Edge*) a vnútorný (*Core*) smerovač. Vnútorný smerovač je smerovač, ktorý smeruje na základe MPLS

návestia. Hraničný smerovač, okrem smerovania na základe MPLS návestia, musí ešte prekladať z pôvodnej technológie z nie MPLS sietí na MPLS návestia, ktoré chce ďalej smerovať do MPLS siete poskytovateľa. Ukážku základného konceptu topológie MPLS siete vidno na obr. 9.4.



Obr. 9.4: Architektúra MPLS siete s vnútornými a hraničnými smerovačmi [27]

Ako na obrázku 9.4 vidieť, hraničné smerovače pridávajú do rámca dodatočne MPLS návestie. Na základe tohto MPLS návestia vnútorné smerovače smerujú. Výstup z MPLS siete opäť zabezpečuje hraničný smerovač, ktorý MPLS návestie z rámca odstráni a umožní tak komunikovať s nie MPLS sieťou.

10 MPLS Traffic Engineering

MPLS *Traffic Engineering* (MPLS TE) je jeden z najmodernejších a v súčasnosti najžiadanejších prostriedkov v sieťach u poskytovateľov služieb. Použitie MPLS v sieťach sa zvyšuje predovšetkým vďaka schopnostiam MPLS TE. MPLS využíva dostupné informácie poskytované smerovacími protokolmi na sieťovej vrstve a operuje ako ATM sieť na linkovej vrstve. S MPLS sú TE vlastnosti integrované do sieťovej vrstvy, ktoré sú využívané na lepšie riadenie využitia prenosového pásma medzi smerovačmi v SP sieti. Ide o manipuláciu dátových tokov tak, aby vhodne zapadli do povahy siete. V kombinácii s MPLS QoS môžeme efektívnejšie riadiť využiteľnosť pásma a zároveň zlepšiť alebo garantovať kvalitu služieb. Riadenie toku dát zlepšuje kvalitu v sieti a zabezpečuje predikatívne podmienky pre dáta. Využitie riadenia toku dát je jeden z prístupov k dátam a zlepšuje manažment využitia dostupných prostriedkov. Jej jednoduchosť, rýchlosť a použiteľnosť je vhodná aj pri nečakaných udalostiach, ktoré nárazovo menia bežné predpokladané podmienky v sieti, ako napríklad významná športová udalosť, škandál celebrity, nový film, aktualizácie operačných systémov, či nečakaný výpadok veľkej časti siete, poškodenie infraštruktúry a podobne. Do niekoľkých minút je možné revitalizovať sieť a aspoň čiastočne obnoviť jej funkčnosť vhodným presmerovaním dátových tokov.

Zlepšenie využitia prenosového pásma dosiahneme vytváraním TE tunelov medzi smerovačmi. Tunel je virtuálna logická cesta od počiatočného bodu ku koncovému. Tunel je jednosmerný a teda koncový uzol tunela nemôže byť zároveň počiatočným bodom toho istého tunela. Tunel môže mať predpísanú cestu, alebo nepredpísanú. Možná je aj kombinácia oboch prístupov. Predpísaná cesta znamená, že dáta – pakety, musia nutne daným uzlom prejsť. Ak nie je definovaná cesta, smerovač sa rozhoduje na základe svojej smerovacej tabuľky. Takéto správanie čiastočne pripomína prepínanie okruhov alebo prepínanie buniek v ATM sieti. Ide teda o akúsi kombináciu zasadenú do konceptu MPLS. Vytváraním TE tunelov môžeme rozkladať záťaž a tak lepšie využiť prenosové pásmo. Jednotlivým tunelom môžeme dodatočne zdefinovať ich vlastnosti a parametre, čo enormne zlepšuje flexibilitu a prispôsobivosť. Je možné riadiť napr. cenu tunela, jeho prioritu, cestu, dáta vstupujúce do tunela a ich kvalitu a podobne. TE je stavané tak, aby v prípade výpadku v sieti – konvergencii, bol čas výpadku čo najkratší. Slúžia nato dodatočné techniky ochrany uzla, linky a cesty. Čas konvergenzie a stratovosť paketov sú omnoho nižšie ako pri použití akejkoľvek inej techniky, či optimalizácie.

Pred vznikom MPLS TE sa v IP sieťach riadil tok dát zmenami parametrov liniek a teda sa zmenil výpočet optimálnej cesty v smerovacích protokoloch. V tomto spôsobe absentuje škálovateľnosť a ovplyvnené sú všetky dátové toky. Dalším spôsobom bolo riadenie politikami (*Policy Based Routing* – PBR). Politika sa definuje na každom zariadení zvlášť. Možnosti a schopnosti sú širšie, no stále nám chýba jednoduchosť riešenia a väčšia škálovateľnosť. V ATM sieťach sa problém rieši úplnou topológiou (*full mesh*), čo však je náročné na manažment a aj pre zariadenia. Výpadok alebo obnovenie znamenajú enormnú záplavu riadiacich informácií, čo vo väčších sieťach spôsobuje vážne problémy s konektivitou. Záťaž je na úrovni $O(N^2)$ až $O(N^3)$ [30]. MPLS TE sa zaraďuje do kategórie CBR (*Constraint Based Routing*), čiže smerovanie na základe požiadaviek.

10.1 Využitie MPLS TE

MPLS TE sa využíva predovšetkým na [30]:

- optimalizáciu využitia prenosového pásma riadenia toku dát
- riešenie nečakaného zahltenia
- riešenie výpadkov uzlov a liniek

Aby sme mohli vôbec využívať funkcie TE, musíme ako smerovací protokol použiť jeden z LS protokolov. V súčasnosti sú len dva LS protokoly s podporou MPLS TE a to OSPF a IS-IS. IS-IS je preferovanejším protokolom, keďže protokol OSPF nepodporuje všetky funkcie, ktoré podporuje protokol IS-IS. IS-IS má už vo svojej podstate a architektúre lepšie predpoklady na implementáciu do IP siete. Je to predovšetkým robustnosť, škálovateľnosť, hierarchický návrh, možnosti optimalizácie, prispôsobivosť a určovanie hraníc oblastí na linkách a nie na smerovačoch ako v prípade OSPF. MPLS TE pridáva aspoň jedno návěstie navyše na jeden tunel do zásobníka návěstí.

MPLS TE používa protokol RSVP pre signalizáciu požiadaviek na jednotlivé tunely. Tento protokol využíva model *IntServ* v kvalite služieb. Nutnou podmienkou pre MPLS TE je aktivovaná technológia MPLS. V prostredí MPLS sa výmena a šírenie návěstí deje najmä prostredníctvom protokolu LDP.

V nasledujúcich častiach popíšeme zaujímavé možnosti MPLS TE.

10.1.1 Atribútové bity

MPLS TE si dokáže vybrať cestu na základe ich vlastností. Máme k dispozícii atribútové bity (*Attribute Flags*), ktoré identifikujú nami definované parametre a vlastnosti linky. Ide o 32-bitovú bitmapu. Pri výbere uzlov a liniek, ktorými bude tunel prechádzať, sa môže tunel rozhodovať na základe týchto vlastností, alebo ich môže aj ignorovať. Kombinovaný prístup je možný, teda časť bitovej mapy je požadovaná a časť je pre tunel nezaujímavá. To, ktoré bity majú byť ako nastavené a ktoré sú sledované, zadávame pri špecifikácii tunela. Nastavujeme takzvanú afinitu (požadované bity) a masku (sledované atribútové bity). Takto je možné riadiť napr. cenu tunela, predpokladané oneskorenie a podobne. Napríklad hlasovým dátam vytvoríme tunel, ktorý nesmie prechádzať cez satelitnú linku, pričom zavedieme atribútový bit pre satelitné linky a nastavíme afinitu a masku tak, aby bol bit v atribútoch liniek sledovaný. Tunel automaticky vyberie optimálnu cestu bez použitia satelitnej linky. Samozrejme, je nutné administratívne každej linke zdefinovať atribúty a zjednotiť ich význam.

10.1.2 Administratívna váha

Zaujímavá je funkcia administratívnej váhy (*Administrative Weight*). Ide v podstate o metriku v MPLS TE, ktorá je implicitne zhodná s metriku IGP protokolu. Jej zmenou získavame ďalší parameter linky. Protokol OSPF vypočíta metriku cesty len na základe rýchlostí liniek, narozdiel od protokolu EIGRP, ktorý počíta aj s oneskoreniami liniek. Protokol IS-IS v mnohých implementáciách rozoznáva hodnotu len na základe jedného parametra, najčastejšie počet skokov. Preto môžeme administratívnu váhu využiť ako ďalší parameter cesty – oneskorenie. Následne zdefinujeme, či sa má tunel rozhodovať na základe najmenšej ceny IGP protokolu – štandardne rýchlosť liniek, počet skokov, alebo na základe administratívnej váhy, čo bude najmenší súčet

oneskorení na linkách. Táto funkcia nám umožňuje diferenciaciu dátových tokov a zlepšuje proces rozhodovania sa pri výbere optimálnej cesty [30]. Ak máme dátové toky, ktoré požadujú nízke oneskorenie a v zásade prenosové pásmo nie je problém, je možné sa rozhodnúť podľa administratívnej váhy a nie IGP metriky. To môžeme použiť napríklad pri VoIP.

10.1.3 Manažment šírenia stavových informácií

Keďže sa predpokladá, že vytvorených tunelov bude násobne viac ako liniek v skutočnosti, tak sa zaviedol manažment šírenia stavových informácií. V prípade výpadku alebo obnovenia linky a uzla sa informácia šíri samozrejme okamžite. Základné pravidlo znie, že konvergencia musí byť čo najrýchlejšia a preto sa tieto informácie nepozdržujú. Inak je to s fázou výpočtu v konvergencii. Zmena na jednej linke môže vyvolať zmeny v desiatkách tunelov vo viacerých zariadeniach naraz. Každý tunel by si musel autonómne vypočítať nové hodnoty a tieto informácie prešíriť do siete, čo je veľmi vysoká záťaž. Preto sa zaviedlo rozdelenie na významné a nevýznamné zmeny. Významná zmena je každá veľká zmena a je vypočítaná bez pozdržania. Najčastejšie ide o relatívne veľký skok v rezervácii prenosového pásma (asociované s protokolom RSVP) a už spomenuté výpadky. Nevýznamné správy sú napríklad relatívne malé zmeny v rezervácii prenosového pásma. Tie sú posielané v periodických intervaloch. Sú zadefinované isté medze, ktoré keď sa presiahnu, tak i nevýznamné zmeny sa stanú významnými. Ešte existuje jeden typ správ a to chybové správy, ktoré sú považované za významné. Zmeny medzi sú konfigurovateľné a taktiež definovanie periodického posielania nevýznamných správ [30].

10.1.4 Opätovná optimalizácia tunelov

MPLS TE ponúka aj možnosť opätovnej optimalizácie tunelov. Ak je tunel aktívny a nastala zmena niekde v sieti, tunel túto zmenu môže zachytiť, ale nereflektuje ju ihneď. Ide častokrát o drobné konfiguračné zmeny ako zmena ceny linky. Prepočítavanie je periodické a možné optimalizácie. Takisto je možné nadiktovať ignorovať časovač a vykonať prepočítanie ihneď. A samozrejme je možné nakonfigurovať udalosti, na základe ktorých sa tento časovač ignoruje. Nechýba ani možnosť zablokovania opätovnej optimalizácie v akomkoľvek prípade, pokiaľ je tunel aktívny. Táto funkcia je dôležitá, lebo zmena po opätovnej optimalizácii môže viesť k zmene cesty, čo spôsobí zmenu oneskorenia v už aktívnej komunikácii medzi koncovými zariadeniami [30]. Takáto zmena je citelná najmä pri VoIP.

10.1.5 Rozkladanie záťaže

Rovnomerné rozkladanie záťaže a taktiež nerovnomerné rozkladanie záťaže je podporované v MPLS TE. Ide o jednu z kľúčových vlastností tejto technológie. Je možné vytvoriť viacero rôznych tunelov medzi zdrojovým uzlom a destináciou s rôznymi metrikami, pričom je možné zadefinovať, aby boli všetky tieto tunely využívané. To, že aká pomerná časť dátového toku bude tiecť tým ktorým tunelom sa definuje na jednotlivých tuneloch administratívne. Aby sme využili túto funkciu naplno, treba mať TE tunel destinácie pozdĺž všetkých ciest a je treba mať zadefinovanú nenulovú hodnotu šírky prenosového pásma. Vzniká otázka, či rozkladanie záťaže neovplyvní negatívne oneskorenie v dátovom toku. Odpoveď znie: nemusí. Závislosť je od implementovaného algoritmu rozkladania záťaže a na základe akých parametrov sa záťaž rozkladá [30]. V Cisco zariadeniach je implementovaný CEF (*Cisco Express Forwarding*), ktorý rozkladá záťaž na základe zadefinovaných parametrov. Avšak pre pakety z jedného dátového toku je

výsledok vždy rovnaký a teda dáta prechádzajú vždy tým istým tunelom. Tu nám nevznikne problém s variabilným oneskorením medzi paketmi, avšak za cenu menej efektívneho rozkladania záťaže. Vhodne zvoliť kritéria pre CEF je nutné vykonať na základe analýzy dátových tokov v sieti.

10.1.6 Posunutie susedstva

Hlavným problémom TE je množstvo vytváraných tunelov. Ak chceme spojiť každý smerovač s každým, hoc len v okrajových bodoch siete, ich množstvo je veľké. Ak uvážime, že občas je nutné vytvoriť pre tú istú cestu viac rôznych tunelov, výsledné číslo je zrazu enormné. V prípade 100 smerovačov máme až 9900 tunelov pri spojení každého s každým, pretože tunel je jednosmerný. A to uvažujeme len jeden tunel pre jednu cestu. Zníženie počtu tunelov je možné pomocou funkcie posunutia susedstva - *MPLS Forwarding Adjacency*. Táto funkcia posúva TE tunel v hierarchii o jednu úroveň bližšie k jadru a teda klesá počet potrebných tunelov pri spojení každého s každým, alebo jedného s viacerými v rámci okrajového bodu siete. Samozrejme, toto posunutie v hierarchii je treba oznámiť smerovačom v nižšej úrovni. Táto funkcia redukuje počet potrebných tunelov. Treba si však dávať pozor, aby ceny tunelov a liniek v IGP pri posunutí boli konzistentné a aby sa rozkladala záťaž, ak je to požadované.

10.1.7 Automatické určenie šírky pásma

Je nutné na každej linke určiť, aká je jej skutočná šírka pásma a v TE tuneloch treba určiť, akú časť tohto pásma môže tunel využiť v prípade potreby. Funkcia automatického určovania šírky pásma (*autobandwidth*) periodicky kontroluje využitie poskytnutého pásma a signalizované požiadavky na pásmo. Smerovač sa môže rozhodnúť, že zmení aktuálne hodnoty prideleného pásma na základe reálnych požiadaviek alebo aktuálnych potrieb pre dátové toky. Tunely tak automaticky zreorganizuje a zmení im hodnoty pridelenej šírky pásma. Druhou možnosťou riadenia je monitorovací nástroj, ktorý vyhodnotí situáciu v sieti a navrhne zmeny, prípadne ich aj aplikuje [30].

10.1.8 Ochrana

V prípade zmeny pomerov v sieti nastáva konvergencia a tá trvá istý čas. V klasickej IP sieti s použitím LS smerovacieho protokolu môže byť čas na sieťovej vrstve až do 10 sekúnd, ak uvažujeme okamžitú detekciu a veľkú sieť. Časy sú konfigurovateľné, no ak sú časy nastavené príliš agresívne, môže dochádzať k falošným správam o zmene a zbytočným konvergenciám. V MPLS TE je každá konvergencia násobne väčšou záťažou. Ak konvergencia nastane na smerovači, ktorý zároveň je východným bodom pre viacero tunelov, tak po skončení výpočtu v smerovacom protokole nastáva konvergencia v tuneloch. Časy konvergence tunelov sú teda dlhšie, keďže sa musí čakať na informácie od smerovacieho protokolu. Zmena časovačov v smerovacích protokoloch nemôže byť nastavená až tak agresívne, ako v prípade nepoužitia MPLS TE. Druhou možnosťou ako zrýchliť konvergenciu tunelov je použitie ochrany.

Ochrana nie je konfigurácia viacerých alternatívnych tunelov, ale konfigurácia tunelov, ktoré sú priamo označené ako záložné. Túto funkciu nazývame aj ako MPLS FRR (*Fast Reroute*) – rýchle presmerovanie. Záložný tunel sleduje primárny tunel a spoločne s ním signalizuje požiadavky. Ak nastane výpadok, záložný tunel môže okamžite po detekcii prebrať úlohu primárneho tunela a tok dát ostáva neporušený. Porušenie je len v čase od výpadku po detekciu, respektíve, výmenu rolí. Je to preto, lebo záložný tunel má už vopred signalizované požiadavky

pozdĺž náhradnej cesty. Alternatívne tunely by po detekcii museli začať signalizovať požiadavky a to samozrejme predĺžuje čas konvergenencie, preto to nepovažujeme za ochranu [RFC4090].

Chrániť môžeme cestu, linku alebo uzol. Podpora ochrany cesty nie je dostatočne implementovaná v dnešných zariadeniach. Ochrana linky znamená, že linku, ktorú chceme chrániť, konfiguráciou záložného tunela obídeme cez iné linky a uzly. Takúto ochranu nazývame taktiež NHop ochrana (*Next-hop Router*). Uzol, na ktorom konfigurujeme záložnú cestu označujeme ako PLR (*Point of Local Repair*). Uzol, na ktorom sa spája záložný tunel s primárnym označujeme ako MP (*Merge Point*). Ochrana uzla znamená vytvorenie záložného tunela, ktorý daný uzol obchádza. Ochrana sa označuje tiež ako NNHop (*Next-next-hop router*).

Ak je nakonfigurovaných viacero záložných TE tunelov na PLR uzle, môže dôjsť k problému výberu optimálnej cesty. Preto sa zaviedla funkcia povýšenia. V periodických intervaloch sa kontrolujú záložné tunely. V prípade, že iný záložný tunel je aktuálne lepší ako momentálny záložný – pripravený prebrať úlohu primárneho, dôjde k výmene rolí a to nazývame povýšením.

Ochrany výrazne skracujú čas konvergenencie, no otázna je ich podpora na zariadeniach. Ako sme si pokusne skúšali, mnohé operačné systémy smerovačov tieto funkcie majú podporované len vo vyšších radách. Ešte existujú ďalšie mechanizmy na fyzickej vrstve, ako napríklad v POS interfejsoch funkcia rýchlej konvergenencie. Síce sa čas konvergenencie udáva do 50 ms, alternatívny tunel musí tak či tak signalizovať svoje požiadavky a čas konvergenencie v TE tuneloch sa nám zlepši len o niečo. Ani ostatné špecifické riešenia ako SONET APS, SDH MSP [31] neriešia hlavný problém tak, ako to rieši ochrana. Je však deklarované, že ochrana – MPLS FRR má lepšie časy konvergenencie ako IP smerovacie protokoly [31, RFC4203, RFC4205]. Toto tvrdenie si prakticky overíme.

10.1.9 Automatická cesta

Automatická cesta (*autoroute*) naviguje IGP smerovací protokol, aby použil MPLS TE tunel ako jeho nasledujúci skok na dosiahnutie koncového uzla a destinácií za týmto uzlom. Výhodou je, že smerovací protokol si nepotrebuje vytvoriť susedstvo cez tunel. Dochádza tak k zmene hodnôt vo fáze výpočtu a teda môže dôjsť k zmene smerovacej tabuľky.

MPLS TE má problém so skupinovou komunikáciou v prípade použitia funkcie automatická cesta. Keďže skupinová komunikácia je smerovaná na základe RPF (*Reverse Path Forwarding*), to znamená, že na základe zdrojovej IP adresy, spomenutá funkcia môže spôsobiť zahadzovanie paketov, keďže v smerovacej tabuľke chýba záznam o takejto ceste. Riešením je nepoužitie danej funkcie, alebo dodatočná aktivácia funkcie, ktorá vytvorí záznam v smerovacej tabuľke tak, ako keby TE tunely ani neexistovali. Podpora je v oboch smerovacích protokoloch OSPF a IS-IS [30].

10.2 Kvalita služieb v MPLS TE

Na riadenie kvality služieb v MPLS TE môžeme použiť všetky známe modely a to *IntServ*, *DiffServ* a *Best Effort*. MPLS TE veľmi dobre spolupracuje s protokolom RSVP, ktorý slúži na signalizáciu požiadaviek dátových tokov v modeli *IntServ*. Tento model má nevýhodu v škálovateľnosti, avšak je možné použiť aj model *DiffServ*. Model *IntServ* nebudeme ďalej analyzovať, keďže sme ho už raz spomínali a v podstate nejde o žiadne zmeny pri použití v MPLS TE sieťach. Jediným rozdielom je, že signalizácia vyberá vhodný TE tunel na splnenie požiadaviek pre dátové toky, CBR.

DiffServ-Aware Traffic Engineering (DS-TE) je spojenie modelu *DiffServ* pre kvalitu služieb a TE [32]. Táto metóda taktiež požaduje protokol RSVP, keďže je súčasťou TE. DS-TE pridáva možnosť diferencovať služby tak, ako to je známe v modeli *DiffServ*. DS-TE pozostáva z piatich súčastí a to [30]:

- Dostupnosť podmnožiny šírky pásma na linke – riadené protokolom RSVP
- Plánovanie na linke – značenie paketov a manažment radov
- Požiadavky na podmnožinu šírky prenosového pásma z východzieho uzla – nároky na TE tunel
- Prístup kontroly tunela z východzieho uzla – administrácia, ktoré a koľko dát vstúpi do TE tunela
- Preempcia tunela – možnosť zostaviť TE tunel pre prioritnejší tok dát a „menej dôležitý“ TE tunel eventuálne aj zrušiť, ak je nedostatok voľnej nerezervovanej šírky pásma

10.3 MPLS TE s MPLS VPN a MPLS QoS

Spojenie MPLS TE s MPLS VPN nepredstavuje žiaden problém. Dokonca je možné pre každého zákazníka vytvoriť hneď niekoľko TE tunelov a diferencovať tak jeho služby, čo však nedáva veľký praktický zmysel. Avšak na okrajových smerovačoch je vhodné zdefinovať niekoľko TE tunelov, ktoré budú prenášať od viacerých zákazníkov ich dáta. Každý tunel bude určený pre iný typ služieb a tak môžeme zdefinovať napríklad osobitný tunel pre VoIP dáta od zákazníkov a osobitné tunely pre ostatné dáta. Spolu s využitím MPLS QoS dosiahneme požadovanú kvalitu služieb a garanciu. A samozrejme, nezáleží na fyzickej infraštruktúre a použitom linkovom protokole, keďže MPLS podporuje mnoho rôznych linkových protokolov, čo sa nazýva AToM (*Any Transport over MPLS*) [30].

11 Protokoly IPv6

Doteraz sme v popise uvažovali len o smerovanom protokole IP verzie 4 - IPv4. Všetky doteraz popísané techniky, prostriedky a ich analýza bola stavaná len pre tento sieťový protokol. Nástupcom protokolu IPv4 je protokol IP verzie 6 - IPv6, ktorý prináša nové možnosti a má viacero výhod. Medzi tie najhlavnejšie by sme zaradili:

- absencia broadcast_u
- využívanie skupinového vysielania
- väčší rozsah IP adres
- globálna dosiahnuteľnosť
- efektívnejšia práca s IP adresami v zmysle sumarizácie na prefixy
- autokonfigurácia
- menšia záťaž na smerovačoch a lepšie podporovaná implementácia pre reťazenie hlavičiek
- roaming a mobilita
- viacdomovci (*multihoming*)
- a mnohé iné

Častokrát sa stretne v literatúre aj s tým, že protokol IPv6 zlepšuje bezpečnosť [33], čo však nie je pravda, keďže neexistuje samostatné pole v hlavičke alebo povinnosť používať protokol zabezpečenia. Avšak protokol je implementačne stavaný tak, že zlepšuje podporu a efektívnosť pri použití zabezpečenia, napr. IPSec.

Protokol IPv6 má však aj mnohé negatíva a to najmä

- väčšia réžia (hlavička je dvojnásobne väčšia)
- neustále zmeny v štandardoch
- ešte stále nedostatočná podpora na strane koncových zariadení, ale aj prepájacích zariadeniach
- absencia aplikácií podporujúcich IPv6

I napriek týmto negatívam však nie je problém už dnes fungovať nad IPv6, avšak je nutné počítať s istými obmedzeniami a nedostatočnou podporou v aplikáciách.

V našom prostredí má však protokol IPv6 obrovské nedostatky. Síce existuje dobrá podpora pre IMS, keďže pôvodný zámer bol postaviť architektúru IMS len nad protokolom IPv6. Avšak absentujú aplikácie, ktoré by nám pomohli verifikovať zvýšenie dostupnosti a kvality služieb. V oblasti smerovania síce existujú smerovacie protokoly pre protokol IPv6 (RIPng, OSPFv3, IS-IS...), avšak ich implementácie sú na úrovni z pred niekoľkými rokmi. To znamená, že nie je možná všetka tá optimalizácia, ktorú potrebujeme na zrýchlenie konvergencie. Nie je možné nastavovať časové parametre s takou granularitou, ako by sme si želali a ako si zaželeli pred niekoľkými rokmi veľkí poskytovatelia služieb. Redundancia cez VRRP protokol, alebo príbuzné protokoly, je tiež nedostatočná, alebo nie je v súčasnosti implementovaná. Úplne najhoršie je, že nie je možné mať MPLS sieť nad protokolom IPv6, keďže nie je implementovaný protokol LDP pre IPv6 a neexistuje ani žiadna alternatíva. Čiže nie je možné využiť MPLS v spojení IPv6 a ani ostatné súčasti MPLS ako MPLS TE [34].

Existuje však jedno riešenie. Ak chce zákazník mať IPv6 sieť, tak poskytovateľ mu to môže umožniť a môže jeho jednotlivé pobočky prepojiť cez svoju IPv4 sieť. Poskytovateľ služieb musí nutne mať IPv4 sieť. Cez protokol MP-BGP (*MultiProtocol BGP*) sa budú vymieňať MPLS návestia. Pre zákazníka je proces transparentný. Tým, že poskytovateľ má IPv4 sieť, je možná optimalizácia, redundancia a MPLS TE v plnom rozsahu, ako bolo analyzované [35].

Po odbornej diskusii sme sa rozhodli, že sa pokúsime našu implementáciu overiť aj nad protokolom IPv6 práve s týmto riešením, keďže je asi jediné relevantné. Ostatné riešenia by si žiadali dodatočnú formu tunelovania, čo ešte viac degraduje naše možnosti a úplne sa odvracia od pôvodného zámeru zvýšenia dostupnosti a použitia MPLS TE.

Otázku IPv6 ponechávame nateraz otvorenú, keďže je problém nájsť materiály k tejto problematike. Ako sme sa dočítali z dokumentácii, novšie verzie implementácií v smerovačoch by už mali podporovať všetku optimalizáciu časových parametrov, ale konkrétnu implementáciu sme zatiaľ nenašli. Našli sme však návrh pre protokol IS-IS, ktorý by mal v budúcnosti podporovať MPLS TE nad IPv6 [36]. Keďže ide len o návrh, ktorý je momentálne diskutovaný, toto riešenie je pre nás momentálne nepoužiteľné. IPv6 postupne doriešime v najbližších týždňoch a aktualizujeme analýzu.

12 Zhodnotenie analýzy

Analýza preukázala, že problematika zvýšenia dostupnosti uzlov a zlepšenia kvality služieb nie je jednoduchou záležitosťou. Postupne sme rozoberali protokol SIP a SDP, architektúru IMS, jej súčasti a vzájomné prepojenia. Do hĺbky sme sa venovali optimalizácii siete najmä na sieťovej vrstve. Popísali sme hlavné smerovacie protokoly a ich techniky, časové a nečasové parametre vhodné na optimalizáciu. Venovali sme sa ich schopnosti dosiahnutia rýchlej konvergenencie. Spomenuli sme všetky ostatné významné techniky pozitívne i negatívne vplyvajúce na konvergenciu. Popísali sme možnosti kvality služieb, spôsoby a prístupy a venovali sme sa technológii MPLS a jej súčasti MPLS TE. Neopomenuli sme ani ani protokol IPv6.

Analýza jasne preukazuje opodstatnenosť práce a dáva tušiť dobré výsledky pri návrhu, v ktorom si kladieme za cieľ optimalizovať sieť na rýchlo konvergujúce bez zníženia ich stability a garantovanie adekvátnej kvality služieb pre aplikácie. Myslíme si, že naše riešenie bude schopné dôstojne konkurovať ostatným prístupom k tejto problematike. Je možné, že aplikovaním všetkých podstatných metód dosiahneme práve my lepšie výsledky. Očakávame totižto synergické efekty po aplikácií relevantných techník.

V nasledujúcej fáze sa zameriame na praktické nasadenie nami popísaných technológií, ich vplyv na sieť a služby a navrhujeme množstvo testov, na ktorých budeme študovať zmeny. Pripravíme niekoľko modelových topológií, na ktorých názorne ukážeme, ako postupovať pri nasadzovaní jednotlivých technológií. Výsledky našej práce podrobne zdokumentujeme.

13 Špecifikácia požiadaviek

V tejto sekcii špecifikujeme hlavné aspekty nášho produktu.

Našou úlohou je zvýšiť dostupnosť jednotlivých uzlov v sieti a najmä kritických uzlov. Za kritický uzol považujeme každý uzol, ktorého nedostupnosť môže spôsobiť finančné straty a nespokojnosť zákazníka.

- Celá študovaná sieť bude pozostávať z dvoch častí: zákazník (zákazníci) a poskytovateľ služieb
- U poskytovateľa služby optimalizujeme počítačovú sieť na dosiahnutie:
 - zvýšenia dostupnosti uzlov a služieb
 - redundancie
 - rýchlej konvergenie pri zmene pomerov v sieti
 - rozloženia záťaže
 - garancie kvality služieb
- Navrhujeme optimalizáciu a skvalitnenie služieb pre koncového zákazníka na strane zákazníka
- Navrhujeme verifikačný postup na overenie skvalitnenia poskytovaných služieb
- Podrobne zdokumentujeme náš postup zabezpečenia skvalitnenia služieb, odporúčania, použité a implementované nástroje a proces verifikácie

Primárne sa budeme snažiť zabezpečiť zvýšenie dostupnosti poskytovaných služieb a zabezpečiť im požadovanú kvalitu. Ako prostriedok plánujeme využiť moderné techniky optimalizácie počítačovej siete, aj na úrovni služieb a aplikácii.

14 Návrh

V tejto časti bližšie popíšeme ako chceme dosiahnuť vytýčené ciele v predošlej kapitole. Testovacie siete zostavíme virtuálne pomocou emulátora Cisco IOS Dynamips/Dynagen a pomocou ostatných virtualizačných nástrojov ako VMware a VirtualBox. Tieto testovacie siete verifikujeme aj na reálnych zariadeniach v našich sieťových laboratóriách, kde budeme simulovať reálnu prevádzku na sieti.

Celý projekt sme rozdelili na dva samostatné celky, ktoré vzájomne prepojíme. Ide o smerovanie a aplikačné služby.

Sieť postavíme na protokole IP a technológii MPLS. Ako médium použijeme *ethernet*. Pri návrhu siete budeme brať do úvahy možnosti v laboratóriách a zároveň použijeme zásady dobrého návrhu ako bolo analyzované. Po konzultácii s odborníkmi z viacerých spoločností poskytujúcich služby si vyberáme smerovací protokol IS-IS. Okrem toho v analýze vyšiel ako najvhodnejší kandidát pre prostredie poskytovateľa služieb.

Na kontrolu a monitorovanie siete využijeme program postavený na protokole SNMP. Zvýšenie dostupnosti uzlov a služieb zabezpečíme pomocou redundancie, čiže využitím VRRP protokolu. Aplikujeme optimalizáciu časových a ostatných parametrov v konfiguračných súboroch smerovačov. V prípade zmeny pomerov v sieti vieme garantovať maximálne dvoj-sekundovú nedostupnosť. Spolu s redundanciou vieme minimalizovať počet kritických výpadkov, kedy sa uzol stane absolútne nedostupným.

Pri MPLS si môžeme vybrať z dvoch základných typov navrhovania MPLS sietí. Ide o taktický a strategický prístup. Pri taktickom návrhu vytvárame TE tunely podľa potreby a pri strategickom návrhu vytvárame v niektorých bodoch siete prepojenie každého s každým. Rozhodli sme sa pre taktický návrh. Vytvoríme niekoľko základných TE tunelov v sieti a budeme púšťať dáta do siete. Budeme postupne zvyšovať nároky na prenosové pásmo, až vznikne potreba nového tunelu alebo presmerovania časti komunikácie. Interaktívne zasiahneme do siete a vytvoríme nový TE tunel. Budeme merať rýchlosť a náročnosť tohto procesu, aby sme vedeli kvantifikovať schopnosť poskytovateľa reagovať na zmeny pomerov v sieti.

Kvalitu služieb nastavíme na strane zákazníka, na strane poskytovateľa služieb, na úrovni prepínania a smerovania a na úrovni aplikácii, čiže naprieč celým spektrom vrstiev RM OSI modelu v celej sieti.

Cieľom implementácie projektu IMS je poskytnúť odberateľom, zákazníkom moderné a technicky vyspelé služby. V dnešnej dobe je prenos hlasu neustále prioritou pri zavádzaní multimediálnych služieb. Neustále však narastajú požiadavky na zložitejšie služby. Pri zavedení systému chceme zákazníkom garantovať nielen prenos v najvyššej možnej kvalite bez citelných obmedzení kvalitou linky a pásma, ale aj vyššie služby ako prenos živého obrazu viacerým klientom. V systéme chceme implementovať aplikačný server schopný prenosu takéhoto vysielania.

Jadro nášho projektu na základe analýzy bude stáť na implementácii IMS OpenIMS Core. Samotný systém IMS bude implementovaný na fyzických výpočtových strojoch. Jednotlivé počítače budú obsluhovať nasledujúce služby CSCF, HSS a AS. Medzi aplikačné servery zahrnieme systémy podporujúce služby ako VoIP a vysielania video záznamu, napríklad programom VLC.

V rámci systému chceme poskytovať niekoľko úrovní kvality našich služieb. Rôzne platiacim zákazníkom budeme garantovať rôznu kvalitu služby. V prípade zlatého zákazníka, zákazník ktorý potrebuje mať garantovanú dostupnosť a kvalitu služieb navzdory zvýšeným nákladom, budú nami poskytované služby mať lepšiu priechodnosť našou transportnou sieťou. Túto službu môžeme garantovať mechanizmom zabezpečenia kvality služieb QoS.

Túto formu služieb chceme zabezpečiť rôznymi médiami bránami v našom systéme. Na ich základe hraničné smerovače budú rôzne označovať pakety RTP prúdu. Zdroj prúdu bude jednotný, ale na základe klasifikácie klienta bude využívať iné média brány.

V prípade úspešného aplikovania smerovacích mechanizmov a mechanizmov garantovania služieb budeme môcť pozorovať rôznu kvalitu vysielaného obrazu u zákazníkov, ktorý požadujú zvýšenú dostupnosť a kvalitu služieb a iných. K tomuto bude môcť dôjsť iba v prípade zahltenia transportnej siete. Ak táto podmienka nebude naplnená klienti nemôžu pociťovať výkyvy v kvalite ponúkaných služieb.

Ako verifikačné nástroje použijeme BASH skripty, ktoré jeden z členov tímu používa na kontrolu rýchlosti konvergencie času, používateľské aplikácie na VoIP hovory, konferenčné hovory a IPTV sledovanie televízie (napr. program VLC), kde si audiovizuálne overíme zmeny v kvalite. Na generovanie dátových tokov si pomôžeme aplikáciami ako napr. PackETH, alebo spustíme FTP prenos a podobne.

Výsledkom bude zvýšenie dostupnosti uzlov a kvality služieb v sieti. Celý progres vytvárania produktu budeme priebežne podrobne dokumentovať. Na záver vytvoríme okrem hlavnej dokumentácie aj technickú dokumentáciu a používateľskú príručku. K projektu priložíme všetky relevantné materiály použité počas tvorby projektu.

V nasledujúcej časti si predstavíme platformu Ericsson SDS na vývoj aplikácií v IMS. Ide o spojenie analýzy možností použitia tohto riešenia s praktickou implementáciou.

14.1 Platforma ESDS na vývoj aplikácií v IMS

Ericsson Service Development, skrátene Ericsson SDS (ESDS), je platforma určená na vývoj aplikácií architektúry IMS v prostredí upraveného nástroja *Eclipse*. Platforma je primárne navrhnutá na fungovanie pod operačným systémom Microsoft Windows. Vďaka zahrnutiu všetkých kľúčových prvkov architektúry IMS Ericsson SDS umožňuje a uľahčuje návrh, vývoj, a testovanie klientských aj serverových aplikácií určených pre nasadenie v systéme IMS [37].

Ericsson SDS v sebe zahŕňa nasledovné prvky siete IMS [37]:

- simulovaný Aplikačný server (*Application server*) – označovaný aj ako tzv. *SIP container*, ktorý umožňuje do IMS siete pridávať služby
- DNS server – umožňuje preklad domén na IP adresy, telefónne čísla či transportné protokoly
- domovský server účastníka (*Home Subscriber Server*) – podporuje poskytovanie údajov, ako sú používateľská identita (*user identity*), prístupové parametre používateľa (*user access parameters*), identity verejných služieb (*Public Service Identities*), a informácie týkajúce sa spúšťania konkrétnych služieb siete IMS (*service-triggering information*)

- *Call session control* funkcia (*Call Session Control Function*) – v prostredí Ericsson SDS sú P-CSCF, I-CSCF a S-CSCF spojené do jedného CSCF prvku
- *Brakeout Gateway Control* funkcia (*Brakeout Gateway Control Function*) – umožňuje simulovať smerovanie hovorov do a z klasických telefónnych sietí

Keďže Ericsson SDS je primárne vývojové prostredie, jeho súčasťou je aj niekoľko knižníc a nástrojov zameraných na vývoj klientskych aplikácií, ako aj aplikácií aplikačného servera. Pre vývoj IMS klientov sa v Ericsson SDS nachádza služba zvaná *Ericsson IMS Platform (ICP - IMS Client Platform)* [37]. Prostredníctvom jej knižnice a API volaní sa z tvorby IMS klienta odstraňuje potreba vytvárania objektov na posielanie, prijímanie a spracovávanie komunikácie s IMS sieťou poskytovateľa. ICP je teda agentom, ktorý sprostredkúva komunikáciu medzi samotným IMS klientom, a sieťou. Pre vývoj programov aplikačného servera je v Ericsson SDS vytvorená podpora pomocou sprievodcov, ktorí uľahčujú vytvorenie skeletu nových aplikácií [38].

Okrem základných prvkov siete IMS sú ešte súčasťou Ericsson SDS aj iné nástroje používané v procese vývoja softvéru na platforme [37]:

- nástroj na vizualizáciu IMS siete (*Visual Network*)
- *IMS Java Client utility* – API, pomocou ktorého je možné vyvíjať IMS aplikácie na mobilné zariadenia s podporou mobilnej verzie Javy
- emulátor prenosných zariadení (*Handheld emulator*) – v ktorom je možné simulovať aplikácie vyvíjané pre zariadenia s *OS Symbian*, napríklad na *Sony Ericsson P1*
- simulátor výskytu manažmentu skupín (*Presence and Group List Management Simulator*)
- simulátor *Push-to-talk* funkcionality v mobilnej sieti (*Push-To-Talk over Cellular Enabler Simulator*)
- simulátor systému okamžitých správ (*IMS Messaging Enabler Simulator*)
- simulátor televízneho vysielania protokolom IP (*IPTV Simulator*)

Platforma Ericsson SDS pre vývojárov predstavuje veľmi užitočný a komplexný nástroj pre vývoj aplikácií pre IMS sieť. Pre vývojárov je k dispozícii kompletná dokumentácia k poskytovaným API v podobe *JavaDoc ALEX* knižnice, nachádzajúcej sa priamo v programe [37]. Bohužiaľ, pri testovaní základnej funkcionality systému sme narazili na niekoľko problémov, ktoré sa nám nepodarilo odstrániť. Systém sme testovali na virtuálnom počítači v programe *VirtualBox*. Hostovaným operačným systémom bolo *Windows XP Profession SP3* legálne získané v rámci MSDNAA dostupnej na FIIT. Testovanou platformou bola Ericsson SDS vo verzii 4.1 FD1 nainštalovaná podľa [39].

Prvým, aj keď menej vážnym nedostatkom, bola absencia balíčka *Microsoft Visual C++ 2005 Redistributable Package* v inštalácii Ericsson SDS, ako aj vo virtuálnom počítači. Dôsledkom absencie týchto knižníc bol fakt, že DNS server dodávaný ako súčasť celej platformy nebolo možné z prostredia programu spustiť. Pridaním balíčka sa však problém odstránil.

Druhým problémom, na ktorý sme počas testovania narazili, sa týkal PGM servera. Na základe postupu popísanom v [40] sme sa snažili spustiť ukážkovú IMS aplikáciu dodávanú spolu s platformou. Po jej úspešnej inštalácii malo nasledovať jej spustenie, avšak chyba v konfigurácii PGM servera nám znemožnila tento krok dokončiť. Chybu sa nepodarilo odstrániť ani po kompletnom reštarte operačného systému a všetkých súvisiacich služieb.

Tretí problém, ktorý sa vyskytol pri testovaní platformy sa týkal prípravy jednoduchého programu aplikačného servera, pričom postup vývoja aplikácie sme získali zo [38]. Tento mal na prijatú SIP správu s metódou *MESSAGE* odosielateľovi odpovedať najskôr správou 200 OK a

potom *MESSAGE SIP* správou obsahujúcou text „*Hello World!*“. Po vložení zdrojových kódov do vygenerovaného skeletu, vygenerovaní nového používateľa a niekoľkých ďalších konfiguračných krokov sa však nepodarilo program na aplikačnom serveri Sailfin v1 spustiť. Pre vylúčenie chyby v konfiguračnom postupe sme platformu Ericsson SDS odinštalovali, počítač reštartovali, a po nasledovnej inštalácii platformy postup znovu zopakovali. Chybu sa nám však odstrániť nepodarilo.

Jediný pozitívny výsledok, ku ktorému sme v rámci testovania platformy Ericsson SDS dospeli, bolo úspešné uskutočnenie hovoru medzi dvoma účastníkmi prihlásenými k tomu istému serveru. Jeden z klientov bol spustený z prostredia virtuálneho počítača a išlo o zdarma dostupnú verziu IMS klienta s názvom Mercurio. Tento mal nastavenú so systémom štandardne dodávanú identitu `alice@ericsson.com`, kde doménu `ericsson.com` hosťovala bežiacia platforma Ericsson SDS. V prípade druhého klienta išlo o program Monster a bol spustený na inom počítači nachádzajúcom sa v lokálnej sieti. Tento program bol do systému prihlásený s taktiež štandardne dodávanou identitou `bob@ericsson.com`. Úspešný hovor bol vykonaný v smere Alica volá Bobovi a došlo k úspešnému prenosu zvuku aj textových správ. Vykonať hovor opačným smerom sa však nepodarilo, ani po výmene klientov na oboch stranách.

15 Prototyp riešenia

V súčasnej fáze projektu sme vytvorili prototyp riešenia, ktorý odráža hrubý návrh nášho riešenia a spĺňa špecifikované požiadavky. Na nasledujúcich stranách popíšeme implementáciu IMS a transportnej siete. Merania špecifikovaných parametrov zatiaľ nevykonáme. Sústreďíme sa na úspešnú implementáciu podľa stanovených kritérií a konfiguráciu komponentov v IMS tak, aby sme v ďalšej fáze projektu mohli pohodlne merať a pokračovať v implementácii podľa potrieb a požiadaviek zákazníka.

V tejto časti postupne rozoberáme jednotlivé komponenty našej IMS siete a uvádzame základný postup inštalácie a konfigurácie. Následne popisujeme navrhnutú transportnú sieť s doteraz aplikovanými technológiami a postup základnej konfigurácie technológií. Optimalizácia transportnej siete bude predmetom neskoršej fázy implementácie, keď budeme vyhodnocovať a porovnávať namerané hodnoty.

Cieľom prototypu riešenia je demonštrovať naše schopnosti praktickej aplikácie analyzovaných oblastí a schopnosť prepojiť svet IMS a transportnú sieť. Návrh zapojenia je stavaný tak, aby bolo možné v druhej fáze projektu pokračovať v implementácii na rovnakej topológii a vyhodnocovať merania.

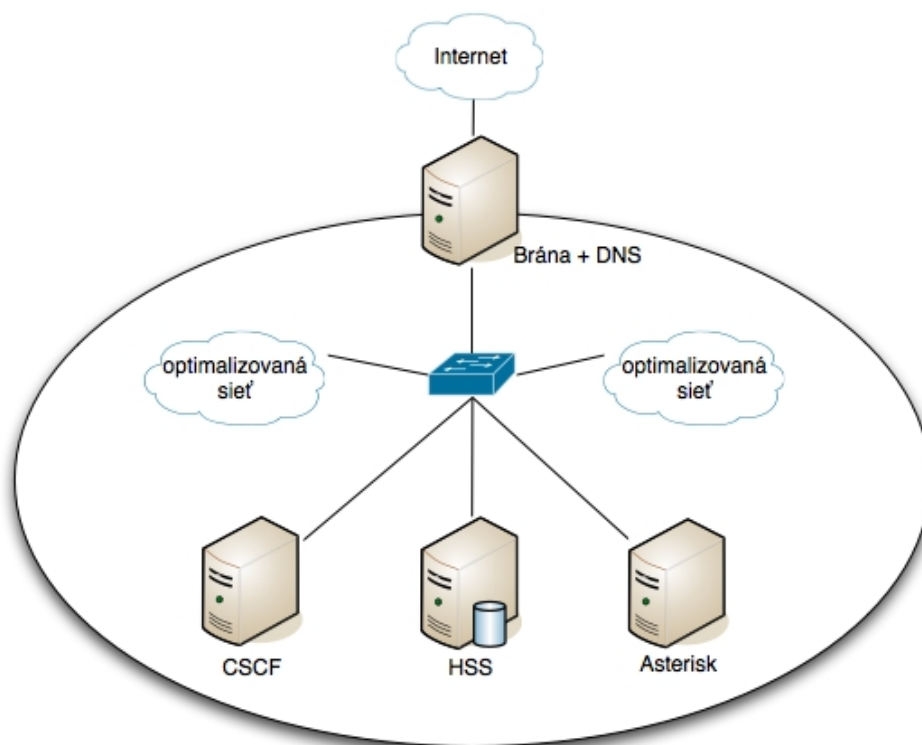
15.1 Topológia v IMS podsieti

Pre nasledovné rozloženie jednotlivých úloh funkcionality IMS sme sa rozhodli na základe analýzy. Z tej vyplynula potreba diverzifikácie výpočtových zdrojov, modularity a modifikovateľnosti komponentov IMS domény. Z dôvodu jednoduchšej údržby a členitosti celkového systému sme jednotlivé funkcie atomizovali do požadovanej miery. Funkcionalitu HSS sme separovali na jediné zariadenie. Takto bude hlavná databáza v prípade potreby aktualizácie, alebo implementovania nových súčastí domény, neustále dostupná. Takisto sa vyhneme problémom v prípade výpadku niektorej súčasti domény. Takto zvyšujeme pravdepodobnosť zachovania dostupnosti služieb.

Na základe výsledkov analýzy navrhujeme topológiu, ktorá sa primárne skladá z nasledujúcich komponentov:

- CSCF – riadiaca jednotka, ktorá umožňuje prístup k doméne a jej službám z hľadiska používateľa.
- HSS – zariadenie s databázou, ktoré je zodpovedné za uchovávanie identity a používateľských informácií jednotlivých používateľov.
- GATEWAY – výpočtové zariadenie, ktoré nám umožňuje vzdialený prístup prostredníctvom protokolu SSH. Administrátorom umožňuje konfiguráciu jednotlivých komponentov architektúry IMS. Tento počítač taktiež poskytuje funkčnosť DNS serveru.
- ASTERISK – systém umožňujúci priame záťažové testovanie základných funkcií VoIP. Systém je použitý iba vo fáze počiatočného testovania. V budúcnosti môže byť použitý ako aplikačný server domény.

Na obrázku číslo 15.1 je znázornená navrhovaná topológia IMS domény začlenená do celkovej architektúry navrhovanej siete.



Obr. 15.1: Topológia navrhovanej IMS domény

V nasledujúcich častiach bližšie popíšeme jednotlivé použité komponenty.

15.1.1 CSCF

CSCF funguje v rámci IMS domény ako riadiaca jednotka zodpovedná za zostavovanie relácií, smerovanie SIP správ a registráciu. Ako sme už uviedli v analýze, poznáme tri typy CSCF:

- *Proxy-CSCF* (P-CSCF)
- *Serving-CSCF* (S-CSCF)
- *Interrogating-CSCF* (I-CSCF)

V architektúre *OpenIMS Core* sú reprezentované ako inštancie programu SER (SIP Express Routes) s rôznymi konfiguračnými súbormi. V našej topológii sú jednotlivé programy spustené na jednom výpočtovom stroji.

Pre inštaláciu danej funkcie sme si zvolili stroj v školskom laboratóriu D-105. Na výpočtový stroj bola nainštalovaná čistá distribúcia operačného systému Linux - Debian Lenny. Nakonfigurované boli najprv sieťové rozhrania. Tak ako všetky súčasti našej domácej domény, výpočtový stroj používa ethernetové rozhranie eth0. Bola mu staticky pridelená IP adresa 5.200.0.2/28 a nastavená predvolená brána pre odchádzajúce dáta a DNS server. Obe hodnoty ukazujú na rovnakú IP adresu 5.200.0.1. Následne sme overili konektivitu komunikáciou s verejnou sieťou.

Podľa postupu zverejnenom vývojármi projektu *OpenIMS Core*, sme zaviedli funkcionality CSCF. Postup inštalácie je nasledovný:

1. Inštalácia operačného systému Debian Lenny so zvolením inštalácie základných balíčkov
2. Príprava zdrojových kódov *OpenIMS Core* systému. Tie sme uložili v adresári */opt/OpenIMSCore/ser_ims* programom *svn* zo systému *Subversion*
svn checkout http://svn.berlios.de/svnroot/repos/openimscore/ser_ims/trunk ser_ims
3. Kompilácia získaných zdrojových kódov programom *make*
make install-libs all
4. Zmena prednastavenej IP adresy. Skriptom *configurator.sh* sme zmenili v konfiguračných súboroch výskyt prednastavenej IP adresy 127.0.0.1 na IP adresu sieťového rozhrania 5.200.0.2.
5. Zavedenie konfiguračných súborov na požadované miesto v adresári */opt/OpenIMSCore*
6. Spustenie inštancie programu SER pomocou skriptov *pcscf.sh*, *icscf.sh* a *scscf.sh*.
7. Po náležitom nastavení funkcionality HSS sme overili funkčnosť konfigurácie domény IMS.

15.1.2 HSS

Táto služba je zodpovedná za uchovávanie a poskytovanie používateľských informácií v rámci IMS siete. Okrem toho je to hlavný prvok zodpovedný za autentifikáciu a autorizáciu používateľov IMS siete. Ako sme už spomenuli, táto funkcia je implementovaná na samostatnom počítači, ktorý je zapojený do spoločnej počítačovej siete.

V rámci architektúry systému OpenIMS, HSS sa skladá z dvoch hlavných častí:

- MySQL databáza
- FHoSS servera

Nakoľko MySQL databáza je z hľadiska IMS siete len úložným priestorom, s ktorým je možné komunikovať iba prostredníctvom MySQL protokolu, je funkcionality HSS doplnená o FHoSS obslužný server, ktorý zastáva úlohu agenta spracúvajúceho jednotlivé požiadavky.

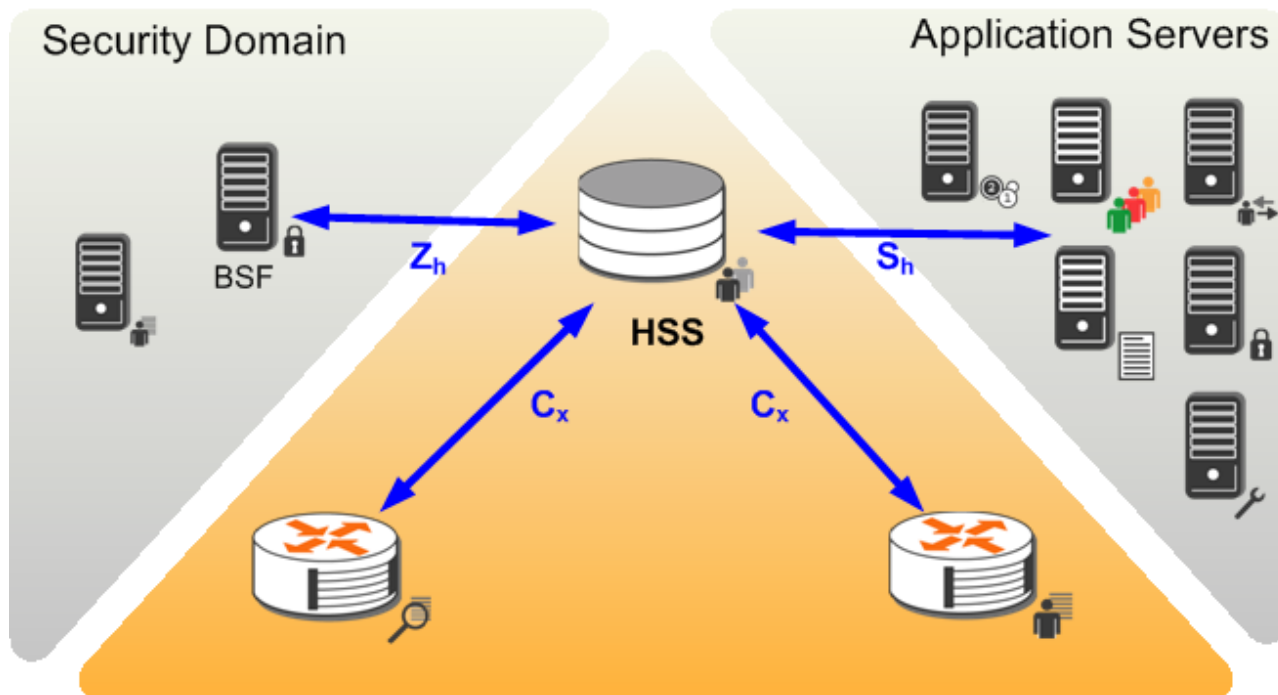
FHoSS server je aplikáciou napísanou v programovacom jazyku Java a z hľadiska architektúry sa skladá z troch vrstiev:

- vrstva sieťového rozhrania (*interface layer*)
- vrstva dátového prístupu (*data access layer*)
- vrstva používateľského rozhrania (*graphical user interface layer*)

Vrstva sieťového rozhrania je zodpovedná za prijímanie požiadaviek a odosielanie odpovedí na jednom z troch dostupných rozhraní, ktoré znázorňuje obrázok č. 15.2.

Vrstva dátového prístupu je postavená na systéme *Hibernate*, teda na implementácii architektúry relačného databázového systému poskytujúceho perzistentné triedy v Jave. Prostredníctvom tejto triedy FHoSS získava údaje z pridelenej MySQL databázy a poskytuje ich zvyšným dvom vrstvám.

Vrstva používateľského rozhrania používa systém *Apache Struts*, teda HTTP server, v ktorom je možné funkcionality web stránok programovať v jazyku Java. Toto rozhranie umožňuje správcovi IMS siete spravovať používateľské údaje, definovať nové účty a iné rôzne operácie spojené s funkcionality HSS uzla.



Obr. 15.2: Rozhrania FHoSS servera [41]

Z technického hľadiska je HSS v rámci nášho projektu implementovaný na bežnom počítači. Nakoľko OpenIMS vyžaduje prostredie operačného systému Linux, bola na tomto uzle použitá minimálna inštalácia distribúcie Debian v najnovšej stabilnej verzii zvanej *Lenny*.

Postup inštalácie HSS uzla bol nasledovný:

1. Inštalácia operačného systému Debian Lenny so zvolením inštalácie základných balíčkov
2. Inštalácia balíčkov potrebných na prevádzku systému FHoSS – *make*, *open-jdk 1.5*, *ant*, *bison*, *flex*, *libxm2* a *openssl*
3. Inštalácia MySQL servera
4. Získanie najnovšej stabilnej verzie FHoSS servera prostredníctvom systému *Subversion* – repozitár sa nachádza na stránke

<http://svn.berlios.de/svnroot/repos/openimscore/FHoSS/trunk>

V čase písania tejto správy tu bola k dispozícii revízia 784.

5. Kompilácia zdrojových kódov FHoSS servera – prostredníctvom programu *ant*.
6. Import predpripravených nastavení systému a používateľov Alice a Bob do MySQL databázy – v našej revízii sa tieto nachádzali v adresári *scripts* a išlo o súbory *hss_db.sql*, *userdata.sql* a *icscf.sql*
7. Spustenie servera, overenie funkcionality

15.1.3 Internetová brána a DNS server

Ďalšie zariadenie v našej sieti slúži ako internetová brána a DNS server zároveň, no budeme ho zjednodušene označovať ako bránu. Funguje na linuxovej distribúcii Debian Lenny a plní v našej sieti tri úlohy:

- dostupnosť vzdialenej správy všetkých počítačov prostredníctvom protokolu SSH

- smerovanie riadiacej prevádzky ostatných uzlov v IMS sieti do siete Internet a preklad adries
- funkcionalita DNS servera v IMS sieti

Nakoľko je testovanie a implementácia IMS siete a optimalizovanej prístupovej siete možné až do fázy merania oddeliť, naša IMS sieť potrebovala počítač, ktorý by bol prostredníctvom vzdialenej správy dostupný z Internetu, a umožnil tak členom tímového projektu s IMS sieťou pracovať aj bez nutnosti fyzickej prítomnosti v laboratóriu. Keďže Internetové pripojenie dostupné v laboratóriu, v ktorom sa tento počítač nachádza, je chránené bránou firewall a s výnimkou tohto stroja sú všetky uzly v IMS sieti pripojené do samostatnej lokálnej siete bez prístupu na Internet, vzdialená správa ostatných prvkov v IMS sieti je možná iba vytváraním SSH spojení z tohto počítača po lokálnej sieti. Tento stroj je zároveň zodpovedný za smerovanie a preklad adries a pre všetky ostatné uzly v IMS sieti slúži zároveň ako predvolená brána pre prístup do siete Internet.

Pre efektívnejšie využitie zdrojov v sieti, ako aj diverzifikáciu služieb s cieľom umožniť ich modularitu bol na tento počítač umiestnený taktiež DNS server, ktorý umožňuje každému prvku v IMS sieti zistiť IP adresy ostatných služieb. Na tento účel sme použili najznámejší DNS server *Bind9*, získaný v najnovšej stabilnej verzii dostupnej z repozitárov distribúcie Debian. Zoznam DNS zón pre tento program sme získali úpravou predpripraveného zoznamu dodávaného so systémom OpenIMS.

Postup inštalácie všetkých služieb na tomto počítači bol nasledovný:

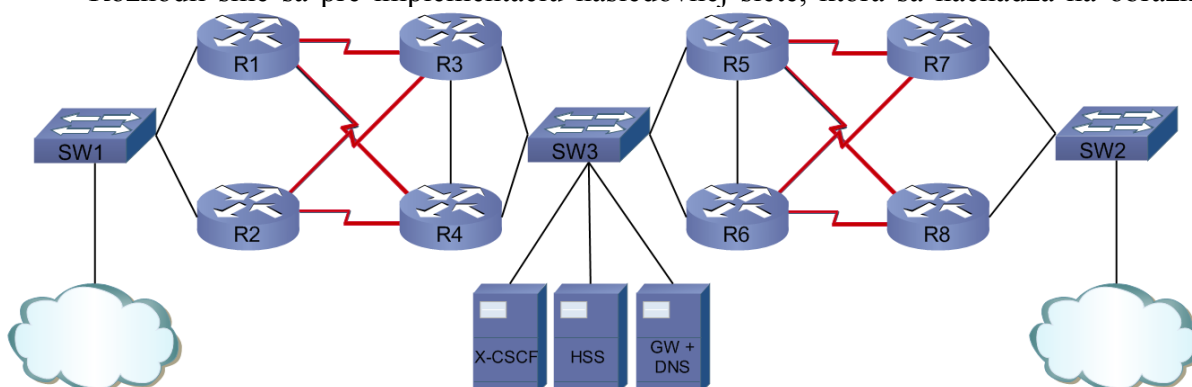
1. Inštalácia operačného systému Debian Lenny so zvolením inštalácie základných balíčkov
2. Inštalácia a konfigurácia brány firewall pre ochranu IMS siete pred nepovoleným prístupom zo siete Internet
3. Konfigurácia smerovania a prekladu adries
4. Inštalácia DNS servera *Bind9* z repozitárov distribúcie Debian Lenny vo verzii 9.5.1
5. Úprava DNS záznamov dodávaných spoločne so systémom OpenIMS a ich nasadenie v DNS serveri
6. Spustenie servera a overenie funkcionality

15.1.4 Asterisk

Za účelom zrýchlenia dostupnosti aspoň minimálnej sady služieb nad protokolom SIP sme vo fáze návrhu na jeden z dostupných počítačov nainštalovali distribúciu Trixbox, ktorá je špeciálne zameraná na poskytnutie služieb softvérovej ústredne Asterisk. Trixbox je postavený na linuxovej distribúcii Centos a okrem samotného programu Asterisk v sebe od inštalácie zahŕňa aj nástroj pre zjednodušenie jeho konfigurácie zvaný FreePBX, ako aj niekoľko ďalších nástrojov, ako správcu odkazovej schránky a podobne. Asterisk sme použili ako dočasnú náhradu za IMS sieť pri testovaní navrhnutého prototypu, nakoľko niektoré súčasti IMS siete ešte v čase testovania neboli v prevádzke. Vo fáze implementácie zadania môže tento server byť použitý ako aplikačný server, prípadne ako iný podporný prostriedok v IMS sieti.

15.2 Transportná sieť

Rozhodli sme sa pre implementáciu nasledovnej siete, ktorá sa nachádza na obrázku 15.3.



Obr. 15.3: Návrh transportnej siete

Cieľom siete je zaistiť dostatočnú redundanciu a rýchlu konvergenciu v prípade výpadku uzla siete alebo liniek medzi uzlami. Smerovače R1 až R4 by bolo možné nahradiť dvoma smerovačmi (napr. R1 a R3). Avšak v prípade výpadku jedného z týchto dvoch uzlov by sa úsek siete stal nedostupným. Redundanciou z obrázka zaistíme, že aj v prípade výpadku viacerých uzlov bude sieť schopná smerovania prevádzky.

Rozhodli sme sa pre dve brány do Internetu, resp. k poskytovateľom služieb – SP (*Service Provider*). Je vylúčené, aby operátor mal iba jeden fyzický prístup do siete WAN. Na obrázku túto časť siete máme znázornenú od SW1 až po „obláčik“. V skutočnosti ide o kritickú časť, kde by pri neželanom výpadku mohlo nastať prerušenie všetkých hovorov operátora.

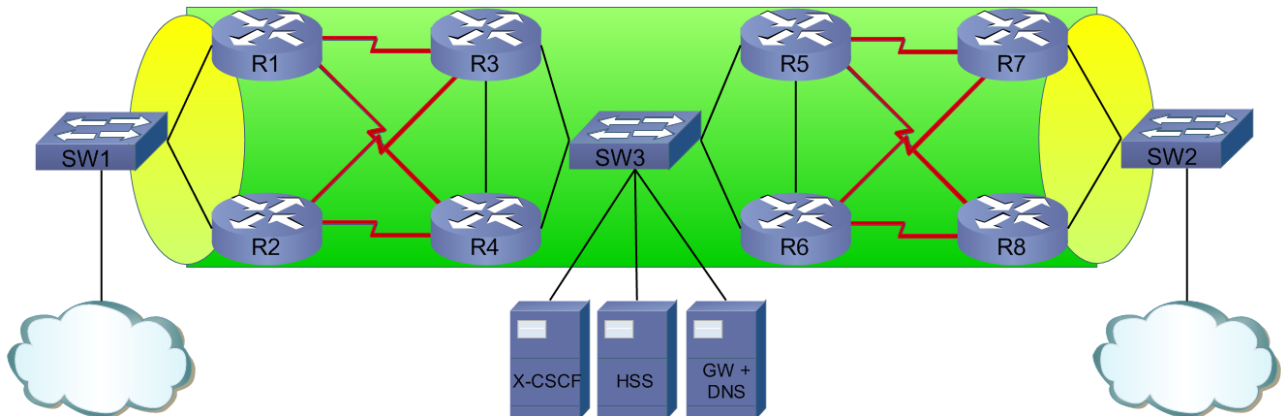
Druhé spojenie do siete WAN je realizované od {R5, R6} až po SW2. Vyťaženie oboch strán bude upresnené pri záťažových testoch. Je možné, že niektoré služby (prípadne používateľov) budeme preferovať jednou cestou a zvyšnú prevádzku budeme smerovať cez druhý prístup k WAN. Každopádne nám redundancia prístupu do WAN rieši problémy s poruchou na strane SP.

V strede topológie siete je jadro prevádzky, ktoré obsahuje uzly so službami. Topológia je navrhnutá univerzálne, teda v jadre môžeme mať ľubovoľný počet rôznych typov služieb.

V prípade implementácie siete operátorom by sme nahradili všetky sériové linky ethernetovými linkami s vysokou šírkou pásma (1 Gbps). V súčasnosti nám laboratórne podmienky zatiaľ neumožnili takúto sieť odsimulovať.

Nazdávame sa, že v našej skúšobnej prevádzke nám pomalšie linky môžu byť prínosom, keďže sme nimi schopní jednoduchšie simulovať správanie sa s preťaženými linkami a tak vyskúšať rôzne techniky QoS.

Na obrázku 15.4 je detailnejšie vidieť VRRP zóny, na ktorých sa uvedené zariadenia budú správať ako jeden logický uzol. Ide o žlté elipsy. Zelená zóna určuje našu sieť a naše zariadenia, ktoré budeme konfigurovať. Do tejto zóny sme zahrnuli len zariadenia, ktoré sú súčasťou transportnej siete.

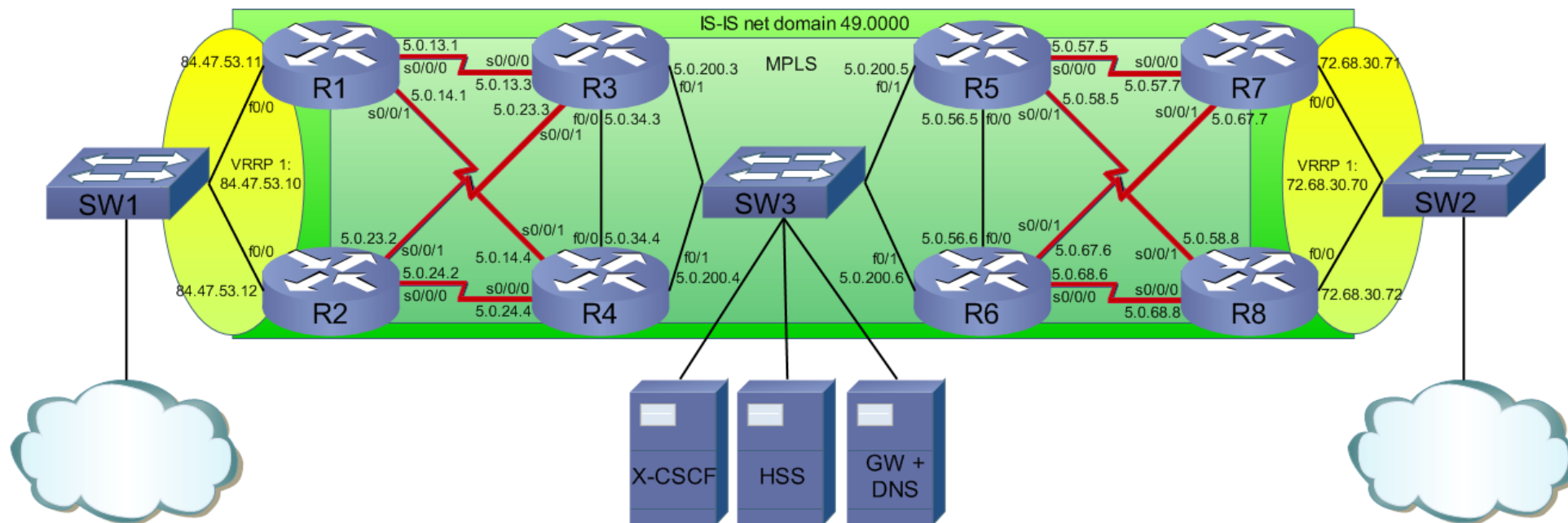


Obr. 15.4: Návrh transportnej siete so zvýraznenými VRRP zónami

Na obrázku 15.5 je vyhotovená celá schéma aj s IP adresami a fyzickým označením interfejsov. Interným sieťovým protokolom bude IS-IS kvôli jeho vysokým možnostiam optimalizácie. Narozdiel od protokolu OSPF ponúka lepšie možnosti optimalizácie, lepšie časy konvergenencie a širšiu paletu možností optimalizácie pri MPLS TE.

V ďalších podkapitolách rozoberieme jednotlivé technológie a obohatíme ich o postup konfigurácie. Postup a konfigurácia sú zamerané na zariadenia od firmy Cisco, keďže sú ako jediné dostupné v laboratóriu. Na prípadné odchýlky u iných výrobcov priebežne upozorníme.

Ukážková konfigurácia smerovačov je prezentovaná tak, ako boli príkazy zadávané do smerovača. To znamená aj príkazy, ktoré sú implicitné a nezobrazujú sa v konfigurácii operujúceho zariadenia.



Obr. 15.5: Detailný návrh transportnej siete

15.2.1 Protokol VRRP

Protokol VRRP ako protokol pre zvýšenie dostupnosti prvého skoku pre koncové zariadenia sme zatiaľ použili na okrajoch siete – smerom k SP. Jeho konfigurácia je v tabuľke 15.1.

Tab. 15.1: Konfigurácia protokolu VRRP

Konfiguračný príkaz	Popis príkazu
Router(config)# interface <i>type number</i>	Nastavenie VRRP na rozhraní smerovača
Router(config-if)# vrrp group ip <i>ipaddress</i>	Nastavenie IP adresy virtuálneho smerovača
Router(config-if)# vrrp group priority <i>level</i>	Zmena priority – úroveň 1-254
Router(config-if)# vrrp group preempt [delay <i>seconds</i>]	Preemptívne vymieňanie si rolí v prípade zmeny priority
Router(config-if)# vrrp group timers advertise [msec] <i>interval</i>	Konfigurácia časovačov

15.2.2 Protokol IS-IS

Prototyp, ktorý bol implementovaný v rámci prvotného návrhu siete, využíva smerovací protokol *Integrated* IS-IS. Tento protokol sme implementovali pre jeho excelentné optimalizačné vlastnosti. Tieto vlastnosti máme vedecky zdokumentované a podložené v [7].

V prototypy bol tento protokol zavedený ako hlavný interný smerovací protokol. Implementovali sme ho s nastavenou sieťovou doménou OSI modelu 49.0000. Číslo 49 označuje privátnu doménu. Zároveň boli v prototypy všetky väzby medzi susednými smerovačmi v rámci IS-IS protokolu ponechané s implicitným nastavením komunikácie. To znamená, že všetky smerovače komunikujú v oboch úrovniach architektúry IS-IS siete.

15.2.3 MPLS

V rámci aktuálneho prototypu bola implementovaná podpora MPLS smerovania. Všetky smerovače (R1 až R8) mali túto podporu aktivovanú. Aktivácia MPLS v prototypy siete vyžaduje len niekoľko splnených požiadaviek. Prvou je podpora CEF (*Cisco Express Forwarding*) na zariadeniach spoločnosti Cisco Systems. Na zariadeniach od iných výrobcov existuje podobná funkcionálna. Druhou je implementovaná podpora pre protokol LDP (*Label Distribution Protocol*) v smerovačoch. Protokol LDP sa aktivuje implicitne po aktivovaní MPLS podpory na danom rozhraní a je potrebný na vybudovanie MPLS smerovania v rámci skonvergovanej topológie. Protokol LDP sme vybrali preto, lebo ide o otvorený štandard. A nezabudli sme zmeniť hodnotu MTU na MPLS aktívnych interfejsoch. Jej hodnota má byť rovnaká alebo väčšia ako typických 1500 bajtov pre ethernet v súčte so všetkými možnými MPLS hlavičkami prechádzajúcimi daným interfejsom.

Konfigurácia protokolu IS-IS a MPLS sa nachádza súhrnne v tabuľke 15.2.

Tab. 15.2: Konfigurácia protokolu IS-IS a MPLS

Konfiguračný príkaz	Popis príkazu
Router(config)# mpls ip	Príkaz aktivuje MPLS smerovanie v rámci IOS smerovača
Router(config)# mpls traffic-engineering tunnels	Príkaz aktivuje pre MPLS smerovania podporu MPLS TE
Router(config)# router isis	Aktivuje IS-IS protokol na danom smerovači a sprístupní jeho konfiguráciu
Router(config-router)# net net_address	Konfigurácia sieťovej adresy v rámci OSI adresácie
Router(config-router)# metric-style wide	Aktivuje rozšírenú metriku v IS-IS
Router(config-router)# mpls traffic-eng router-id interface	Nastavenie identifikátora smerovača v rámci MPLS TE v IS-IS
Router(config-router)# mpls traffic-eng level	Nastavenie prikazuje posielať informácie z MPLS TE pomocou spojení danej úrovne v IS-IS
Router(config-if)# ip router isis	Aktivuje smerovanie v IS-IS na danom rozhraní
Router(config-if)# mpls ip	Aktivuje podporu MPLS pre dané rozhranie
Router(config-if)# mpls mtu bytes	Zmení hodnotu MTU pre interfejs
Router(config-if)# mpls traffic-eng tunnels	Aktivuje podporu MPLS TE pre dané rozhranie
Router(config-if)# ip rsvp bandwidth bandwidth sub-bandwidth sub-bandwidth	Aktivuje RSVP na danom rozhraní a alokuje prenosové pásmo na danom rozhraní

15.3 Zhodnotenie

Podstatou hrubého návrhu a prototypu riešenia bolo vytvorenie modelovej topológie pre IMS sieť a transportnú sieť a vzájomne previazať obe súčasti. V súčasnej fáze implementácie existuje zapojená a funkčná transportná sieť so základnými konfiguračnými príkazmi. Sieť IMS je nakonfigurovaná a napojená do siete tak, aby mohla byť verifikovateľná dostupnosť. Prototyp riešenia svoj cieľ splnil.

V ďalšej fáze projektu budeme optimalizovať transportnú sieť tak, aby sme dosiahli lepšie časy konvergenencie, zvýšili redundanciu a zvýšili dostupnosť uzlov. Vykonáme merania a porovnáme hodnoty s neoptimalizovanou sieťou. Do IMS zavedieme nové servery a nové služby, ako napríklad IPTV a budeme vyhodnocovať ich dostupnosť a kvalitu. V rámci celej siete aplikujeme kvalitu služieb. Celý proces podrobne zdokumentujeme.

Slovník skratiek

ABR - Area Border Routers	IS-IS - Intermediate System to Intermediate System
ADSL - Asymmetric Digital Subscriber Line	ISPF - Incremental SPF
AS - Application Servers	ISUP - ISDN User Part
ASBR - Autonomous System Boundary Routers	KP - Konvergencia protokolu
AToM - Any Transport over MPLS	LLQ - Low-Latency Queuing
BDR - Backup Designated Router	LDP - Label Distribution Protocol
BGCF - Breakout Gateway Control Function	LSA - Link-State Advertisement
BGP - Border Gateway Protocol	LS - Link-State
BICC - Bearer Independent Call Control	LSP - Link State Packet
CBR - Constraint Based Routing	MGCF - Media Gateway Control Function
CBWFQ - Class-Based WFQ	MPLS - Multi-Protocol Label Switching
CEF - Cisco Express Forwarding	MPLS TE - MPLS Traffic Engineering
CIDR - Classless Inter-Domain Routing	MP - Merge Point
CLNS - Connectionless Network Services	MP-BGP - MultiProtocol BGP
CoS - Class of Service	MRFC - Multimedia Resource Function Controller
CQ - Custom Queuing	MRFP - Multimedia Resource Function Processor
CSCF - Call Session Control Function	MTU - Maximum Transmission Unit
CS CN - Circuit Switched Core Network	NAT - Network Address Translation
CSPF - Constrained Shortest Path First	NAT-PT - Network Address Translation - Protocol Translation
DBD - DataBase Description packet	NBMA – Nonbroadcast Multi-Access
DIS - Designated Intermediate System	NGN - New Generation Networks
DNS - Domain Name Server	NHop - Next-hop Router
DR - Designated Router	NNHop - Next-next-hop router
DS-TE - DiffServ-Aware Traffic Engineering	OSI - Open Systems Interconnection
DV - Distance Vector	OSPF - Open Shortest Path First
EBGP - External Border Gateway Protocol	PBR - Policy Based Routing
ERP - Exterior Routing Protocol	P-CSCF - Proxy-CSCF
ESDS - Ericsson Service Development	PDF - Policy Decision Function
ES - End System	PEF - Policy Enforcement Function
FIFO - First In First Out	PLR - Point of Local Repair
FRR - Fast Reroute	POS - Packet Over SONET
GLBP - Gateway Load Balancing Protocol	POTS - Plain Old Telephone Service
HSRP - Hot Standby Router Protocol	PQ - Priority Queuing
HSS - Home Subscriber Server	PRC - Partial Route Calculation
HTTP - Hyper text transport protocol	PSTN - Public Switched Telephone Network
IBGP - Internal Border Gateway Protocol	QoS - Quality of Service
ICP - IMS Client Platform	RED - Random Early Detection
I-CSCF - Interrogating-CSCF	RIP - Routing Information Protocol
IGP - Interior Gateway Protocol	RPF - Reverse Path Forwarding
IMS - IP Multimedia Subsystem	RSVP - Resource Reservation Protocol
IOS - Internetwork Operating System	RTCP - Real-Time Control Protocol
IP - Internet Protocol	RTP - Real-Time Protocol
IRP - Interior Routing Protocol	
ISDN - Integrated Services Digital Network	
IS - Intermediate System	

SCIP - Simple Conference Invitation Protocol
S-CSCF - Serving-CSCF
SDP - Session Description Protocol
SEG - Security Gateway
SER - SIP Express Routes
SGW - Signaling Gateway
SIP - Session Invitation Protocol
SLF - Subscription Locator Function
SP - Service Provider
SPF - Shortest Path First
SS7 - Signaling System 7
TCP - Transmission Control Protocol
THIG - Topology Hiding Inter-network Gateway
TLS - Transport Layer Security
TOS - Type Of Service
UDP - User Datagram Protocol
URI - Uniform Resource Identifier
URL - Uniform Resource Locator
VIP - Virtual IP
VoIP - Voice over IP
VPN - Virtual Private Network
VRID - Virtual Router ID
VRRP - Virtual Router Redundancy Protocol
WFQ - Weighted Fair Queuing
WRED - Weighted RED

Literatúra

- [1] Hersent, Petit, Gurle: IP Telephony - Deploying Voice-over-IP Protocols. John Wiley & Sons Ltd, 2005. 416 s. ISBN-13: 978-0470023594.
- [2] Johnston: SIP: understanding the Session Initiation Protocol. ARTECH HOUSE, 2004. 310 s. ISBN-13: 978-1580536554.
- [3] Camarillo: SIP Demystified. McGraw-Hill Professional, 2001. 320 s. ISBN-13: 978-0071373401.
- [4] KELLY: VoIP for dummies. Wiely Publishng, 2005. 312 s. ISBN-13: 978-0764588433.
- [5] Poikselka, M., Mayer, G., Khartabil, H.: The IMS: IP Multimedia Concepts and Services. John Wiley & Sons Ltd, 2006. 439 s. ISBN-13: 978-0-470-01906-1.
- [6] Russell, T.: The IP Multimedia Subsystem (IMS): Session Control and Other Network Operations. McGraw-Hill Osborne Media, 2008. 224 s. ISBN: 0071488537.
- [7] Burda, F.: Konvergencia v počítačových sieťach. Bakalárska práca, FIIT STU, 2009.
- [8] Reynders, D., Wright, E.: Practical TCP/IP and Ethernet Networking. IDC Technologies, August 2003. 306 s. ISBN 0-7506-5806-1.
- [9] Burda, F.: Convergence in Computer Networks. In: Proceedings of the 5th Student Research Conference in Informatics and Information Technologies. Bratislava, Slovak Republic, 2009. pp. 327-334.
- [10] EIGRP, IP Routing, IGRP. <http://www.rhyshaden.com/eigrp.htm> (28.10.2009)
- [11] BGP Path Selection, <http://rbcciequest.wordpress.com/2008/02/27/bgp-path-selection/> (28.10.2009)
- [12] OSPF routing protocol, dijkstra algorithm, ospf stub area. <http://www.rhyshaden.com/ospf.htm> (28.10.2009)
- [13] Poretzky, S.: Terminology for Benchmarking IGP Data Plane Route Convergence. Internet Draft. Jún 2003. <http://www.ietf.org/proceedings/03jul/I-D/draft-ietf-bmwg-igp-dataplane-conv-term-00.txt> (2008-09-11)
- [14] Dubois, N., Fondeviole, B., Michel, N.: Fast convergence project. France Telecom R&D, 2004.
- [15] Pužmanová, R.: Routing and Switching, Time of Convergence?. Addison-Wesley, 2002. ISBN 0-201-39861-3.
- [16] Daugherty, B.: Optimizations for Routing Protocol Stability and Convergence. Cisco Systems, Inc., 2002.
- [17] Shaikh, A., Greenberg, A.: Experience in Black-box OSPF Measurement. AT&T Research, 1998.
- [18] ISISv6/BGP Fast Convergence Tuning. 6NET Project, 2003.
- [19] Alaettinoglu, C., Jacobson, V., Yu, H.: Toward Millisecond IGP Convergence. Packet Design, Inc., 2000.

- [20] Previdi, S., Horrocks, P.: Integrated IS-IS Design and Deployment Guide, 1998.
- [21] Cisco Systems, Inc.: Cisco IOS Quality of Service Solutions Command Reference, Október 2009, s. 208-350. http://www.cisco.com/en/US/docs/ios/qos/command/reference/qos_cr.pdf (2009-10-26)
- [22] Jork, M., Fang, L.: LDP IGP Synchronization. Internet Draft. December 2008. <http://tools.ietf.org/html/draft-ietf-mpls-ldp-igp-sync-04> (2009-10-26)
- [23] Cisco Systems, Inc.: MPLS LDP Session Protection. Máj 2007. http://www.cisco.com/en/US/docs/ios/12_0s/feature/guide/fssespro.pdf (2009-10-26)
- [24] Cisco Systems Curriculum: Optimizing Converged Networks v5.0. Cisco Networking Academy, 2007.
- [25] Cross V.: Linux on e-server IBM zSeries and S/390: Virtual Router Redundancy Protocol on VM Guest LANs. <http://www.redbooks.ibm.com/redpapers/pdfs/redp3657.pdf> (28.10.2009)
- [26] Cross V.: Linux on e-server IBM zSeries and S/390: Virtual Router Redundancy Protocol on VM Guest LANs. <http://www.redbooks.ibm.com/redpapers/pdfs/redp3657.pdf> (28.10.2009)
- [27] Cisco Systems Curriculum: Implementing Secure Converged Wide-area Networks v5.0. Cisco Networking Academy, 2007.
- [28] Pepelnjak, I., Guichard, J.: MPLS and VPN architectures - volume II. Cisco Press 201 West 103rd Street Indianapolis, IN 46290 USA, 2001.
- [29] Pepelnjak, I., Guichard, J.: MPLS and VPN architectures - volume I. Cisco Press 201 West 103rd Street Indianapolis, IN 46290 USA, 2001.
- [30] Osborne, E., Simha, A.: Traffic Engineering with MPLS. Cisco Press, Júl 2002. 608 s. ISBN 1-58705-031-5.
- [31] Álvarez, S.: MPLS Traffic Engineering Traffic Protection using Fast Re-route (FRR). Cisco Systems, Inc, August 2008.
- [32] Cisco Systems, Inc.: MPLS Traffic Engineering - DiffServ Aware (DS-TE) http://www.cisco.com/en/US/docs/ios/12_2s/feature/guide/fdserv3.pdf (2009-10-17)
- [33] Satrapa, P.: Internetový protokol IPv6. CZ.NIC, z. s. p. o., 2008. ISBN 978-80-904248-0-7.
- [34] 6net, An IPv6 Deployment Guide. The 6NET Consortium, September 2005.
- [35] Polyraakis, A., Kalogeras, D.: 6PE: IPv6 over MPLS. N.T.U.A Network Management Center. CISCO EXPO, Január 2005.
- [36] Bartlett, M., Berger, J., Harrison, J.: IPv6 Traffic Engineering in IS-IS. Internet Draft. September 2009. <http://www.ietf.org/id/draft-ietf-isis-ipv6-te-07.txt> (2009-11-07)
- [37] Service Development Studio (SDS) 4.1 Developer's Guide. Ericsson AB, 2009. 418s.
- [38] Service Development Studio (SDS) 4.1 Tutorial. Ericsson AB, 2009. 232s.
- [39] Service Development Studio (SDS) 4.1 FD1 Installation Instructions. Ericsson AB, 2009. 102s.
- [40] Service Development Studio (SDS) 4.1 Windows Client Sample Application Description. Ericsson AB, 2009. 54s.

- [41] Dokumentácia pre Open IMS HSS implementáciu.
<http://www.openimscore.org/docs/FHoSS/index.html> (2009-12-06)

Relevantné RFC dokumenty

- [RFC142] Kline, Ch., Wong, J.: Time-Out Mechanism in the Host-Host Protocol. RFC142, Máj 1971.
- [RFC793] Information Sciences Institute University of Southern California. Transmission Control Protocol. RFC793, September 1981.
- [RFC1142] Oran, D.: OSI IS-IS Intra-domain Routing Protocol. RFC1142, Február 1990.
- [RFC2327] Handley, M., Jacobson, V.: SDP: Session Description Protocol. RFC2327, Apríl 1998.
- [RFC2328] Moy, J.: OSPF Version 2. RFC2328, Apríl 1998.
- [RFC2338] Knight, S., Weaver, D., Whipple, D.: Virtual Router Redundancy Protocol. RFC2338, Apríl 1998.
- [RFC2396] Berners-Lee, T., Fielding, R., Masinter, L.: Uniform Resource Identifiers (URI): Generic Syntax. RFC2396, August 1998.
- [RFC2543] Handley, M., Schooler, E., Schulzrinne, H.: SIP: Session Initiation Protocol. RFC2543, Marec 1999.
- [RFC3031] Rosen, E., Viswanathan, A., Callon, R.: Multiprotocol Label Switching Architecture. RFC3031, Január 2001.
- [RFC3101] Murphy, P.: The OSPF Not-So-Stubby Area (NSSA) Option. RFC3101, Január 2003.
- [RFC3261] Camarillo, G., Rosenberg, J., Schulzrinne, H.: SIP: Session Initiation Protocol. RFC3261, Jún 2002.
- [RFC4090] Pan, P., Swallow, G., Atlas, A.: Fast Reroute Extensions to RSVP-TE for LSP Tunnels. RFC4090, Máj 2005.
- [RFC4203] Kompella, K., Rekhter, Y.: OSPF Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS). RFC4203, Október 2005.
- [RFC4205] Kompella, K., Rekhter, Y.: Intermediate System to Intermediate System (IS-IS) Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS). RFC4205, Október 2005.
- [RFC4271] Hares, S., Li, T., Rekhter, Y.: A Border Gateway Protocol 4 (BGP-4). RFC4271, Január 2006.