ŽILINSKÁ UNIVERZITA V ŽILINE
FAKULTA RIADENIA A INFORMATIKY

DIPLOMOVÁ PRÁCA

Študijný odbor: Informačné systémy

Bc. Peter Smítka

Tvorba QoS architektúr
Pre IP telefóniu

Vedúci: Ing. Pavel Segeč, PhD.

Reg. č.: 146/2006 Máj 2007

Veľký diel
010 26 ŽILINA

Anotačný list

RESUMÉ

Diplomová práca sa zaoberá QoS sieťovými mechanizmami na zabezpečenie kvality IP telefónnych hovorov v zmiešaných sieťach. Popisuje možnosti zabezpečenia, ich praktickú konfiguráciu a taktiež testuje ich účinnosť na testovacej sieti zostavenej v laboratóriu.

SUMMARY

This diploma thesis is dedicated to study QoS network mechanisms for ensuring the quality of IP telephony calls in mixed networks. It describes the possibilities, configuration and test the mechanisms efficiency in laboratory conditions.

Čestné prehlásenie

Prehlasujem, že som diplomovú prácu spracoval samostatne a že som uviedol všetky použité pramene a literatúru, z ktorých som čerpal.

V Žiline 18.mája 2006

Poďakovanie

Touto cestou ďakujem vedúcemu diplomovej práce, Ing. Pavlovi Segečovi, PhD. za jeho rady, konštruktívnu kritiku a čas, ktorý mi venoval pri konzultáciách. Taktiež chcem poďakovať Ing. Petrovi Palúchovi za jeho rady pri praktickej časti a jeho čas.

1. Úvod 3
2. Voice over IP (VoIP) 4
2.1 Real-Time Transport Protokol (RTP) 5
2.2 RTP Control Protokol (RTCP) 7
2.3 Signalizácia 7
2.3.1 H.323 7
2.3.2 Session Initiation Protocol (SIP) 8
2.4 Audiokodeky 17
2.5 Meranie kvality hlasu - MOS 19
3. Quality of Service (QoS) 23
3.1 Čo je QoS 23
3.2 Faktory ovplyvňujúce VoIP QoS 24
3.3 Strata a oneskorenie paketov 26
3.3.1 Typy sieťového oneskorenia 26
3.4 Metódy implementácie QoS 29
3.4.1 IntServ Model 29
3.4.2 DiffServ Model 31
3.4 QoS metódy špecifické pre VoIP 33
3.5 Komponenty QoS pre VoIP 34
3.5.1 Kompresia 35
3.5.2 Marking 36
3.5.3 Queuing 38
3.5.4 Traffic Shaping a Policing 44
3.5.5 Fragmentácia 45
3.6 Link efficiency mechanizmy v Cisco IOS 47
3.7 QoS a VoIP signalizácia 48
4. Použité zariadenia a softvérové prostriedky 50
4.1 Model siete 50
4.2 Použité zariadenia 52
4.3 Použité meracie prostriedky 53
5. Konfigurácia QoS mechanizmov v IOS 61
5.1 Všeobecná konfigurácia Cisco IOS 61
5.2 Inštalačné skripty 63
5.2.1 Konfigurácia LLQ a IP RTP Priority 63
5.2.2 LLQ 64
5.2.3 IP RTP Priority 66
5.2.4 Link Fragmentation a Interleaving : Multilink PPP 66
5.2.5 Compressed Real-time Protocol (cRTP) 66
5.2.6 Auto QoS 67
6. Testovacie scenáre a výsledky testov 68
6.1 Testy bez použitia QoS 69
6.1.1 Scenár 1 – Bez QoS a bez záťaže 69
6.1.2 Scenár 2 – Bez QoS a so záťažou 71
6.2 Testy s použitím QoS 73
6.2.1 Scenár 3 – Low Latency Queuing (LLQ) 73
6.2.2 Scenár 4 – Low Latency Queuing (LLQ) s LFI a cRTP 75
6.2.3 Scenár 5 – IP RTP Priority 76
7. Hodnotenie a záver ....- 79 -
8. Zoznam skratiek ....- 80 -
9. Prílohy ....- 83 -
9.1 Konfigurácia klasifikácie smerovačov R1 a R5 ....- 83 -
9.2 Konfigurácia prepínača napojeného na VQManager ....- 84 -
9.3 Konfigurácia smerovača R3 ....- 85 -
9.3.1 – R3 s LLQ ....- 85 -
9.3.2 – R3 s LFI, LLQ a cRTP ....- 87 -
9.3.3 – R3 s IP RTP Priority ....- 89 -

9.4 Konfigurácia smerovača R4 ..... - 91 -
9.4.1 – R4 s LLQ ..... - 91 -
9.4.2 – R4 s LFI, LLQ a cRTP ..... - 93 -
9.4.3 – R4 s IP RTP Priority ..... - 95 -
9.5 Schéma testovacej siete ..... - 97 -

1. Úvod

V dnešnej dobe, keď sa v telekomunikačnom prostredí vo veľkom nasadzujú riešenia konvergovaných sietí zlučujúcich dáta, hlas, video do jedného logického paketového celku, je nutné sa zamyslieť nad kvalitatívnymi požiadavkami týchto jednotlivých zložiek. Prenos hlasu cez IP sieť, skrátene VoIP (Voice over Internet Protocol) v dnešnej dobe získava stále viacej na dôležitosti. V dobe relatívne vysokej penetrácie DSL (Digital Subscriber Line) prípojok, a iných prístupových metód do siete internet, prichádza čoraz viac tradičných aj alternatívnych operátorov s ponukou VoIP ako alternatívou k bežnej PSTN (Public Switched Telephone Network) sieti.
VoIP je jednoznačne technológiou blízkej budúcnosti a o jej úspechu sa nedá pochybovať. Predsa však existujú pri VoIP isté predpoklady a požiadavky, ktoré musia byť splnené. Na to aby bola zaručená požadovaná kvalita služby, tzv. QoS (Quality of Service), existuje nepreberné množstvo techník a konceptov.

V rámci tejto diplomovej práce boli v RCNA laboratóriu sieťovej akadémie Cisco analyzované a vykonané testy s vybranými QoS mechanizmami so zreteľom na prenos hlasu po IP sieti. Testy boli vykonané na vopred navrhnutej testovacej zostave pozostávajúcej zo zariadení Cisco (smerovače, prepínače), využíval som tak aktívne ako aj pasívne softvérové prvky pre meranie hodnôt QoS a na simuláciu VoIP hovorov boli použité tak reálne VoIP telefóny (softvérové aj hardvérové) ako aj simulačné programy pre generovanie a vyhodnocovanie jednotlivých parametrov hovorov.

Cieľom tejto diplomovej práce je analýza a návrh optimálneho riešenia prenosu VoIP hovorov v prostredí sietí LAN a čiastočne WAN liniek za použitia dnes dostupných metód a nástrojov. K tomuto účelu boli vytvorené sady štatistík z meraní, pomocou ktorých je možné vybrať to konkrétne riešenie, ktoré najviac vyhovuje danej implementácii.

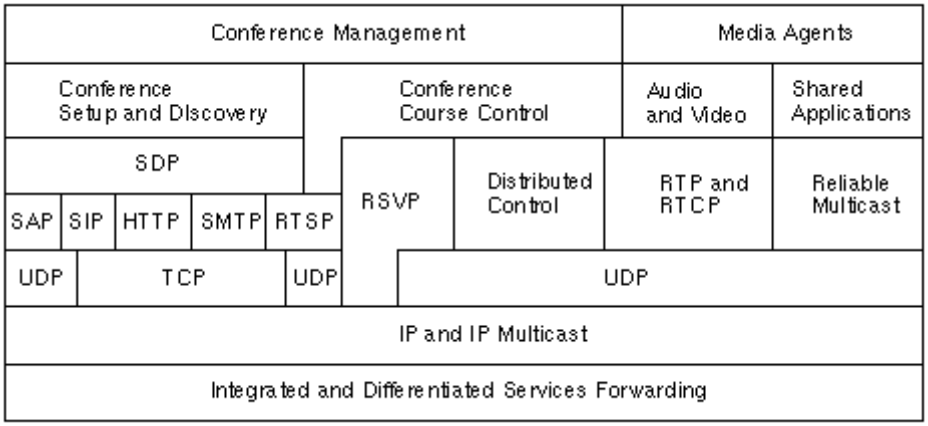
2. Voice over IP (VoIP)

Väčšina dnešných existujúcich telefónnych sietí založená na princípe PSTN (Public Switched Telephone Network) sietí. To znamená, že hovor si rezervuje spojenie medzi dvoma užívateľmi a nik iný toto spojenie využívať nemôže. Keď sa hovor ukončí, linka sa uvoľní a je voľná pre ostatných užívateľov.
Rozdiel v Internetovom telefonovaní, tiež známom ako Voice-over-IP (VoIP) alebo IP telefónia (IPtel) je, že prenos je uskutočnený v rámci IP paketovej siete. Tu je možné posielať pakety medzi dvoma a viacerými stranami bez rezervácie spojenia. Dosiahneme to digitalizáciou audio signálov, zaznamenaných použitím mikrofónu, IP telefónu alebo počítača, zapuzdrením do paketov a potom poslaním po sieti použitím internetových protokolov. Druhá strana potom dané pakety poskladá, dekapsuluje a prehrá pôvodnú správu pomocou reproduktorov. Iné typy médií ako napríklad video alebo zdieľané aplikácie je tiež možné zahrnúť do prenosu bez výraznejších zásahov a zmien. Temnou stránkou veci tejto metódy komunikácie je náročnosť zabezpečenia kvality služby tiež známej pod skratkou QoS (Quality of service), keďže nie je žiadny spôsob ako garantovať že paket dorazí do jeho cieľa. Tento problém by sa dal minimalizovať nepoužívaním verejnej internetovej siete ale radšej privátnych menežovateľných sietí, kde sa dajú isté aspekty QoS garantovať.
Ku komunikácii cez VoIP môžeme použiť buď špeciálne hardvérové IP telefóny alebo VoIP softvér ako napr. X-Lite alebo SJPhone.

Obr. 2.1 – 1 Príklad softvérového a hardvérového VoIP telefónu

Požiadavka funkcie signalizácie odlišuje Internetovú telefóniu od iných internetových multimediálnych služieb ako je real-time vysielanie (real time media broadcasting) alebo služby na vyžiadanie. Táto signalizácia musí byť schopná vytvárať, organizovať a riadiť hovory. Jedným z problémov Internetovej telefónie je lokalizácia volajúcich účastníkov telefónneho hovoru. Osobná mobilita, zisťovanie dostupnosti a iné faktory robia proces signalizácie oveľa zložitejším. Pre túto úlohu je možné použiť Session Initiation Protocol (SIP), ktorý je súčasťou súboru protokolov vytvorených organizáciou IETF (Internet Engineering Task Force) nazvanom - Internet Multimedia Conferencing Architecture. SIP protokol prekladá adresy aplikačnej vrstvy, zahajuje a spravuje hovory. Iný protokol alebo súbor protokolov, ktorý vznikol v ITU (International Telecommunication Union) je H.323 ktorý sa funkcionalitou podobá protokolu SIP.
Jednou z najväčších výhod Internetovej telefónie je, že nie je obmedzená na jeden druh média a linky alebo unicastovému prenosu. V porovnaní s PSTN je výhodou

transparentnosť siete a prenášaného média, tak, že pridanie ďalšieho druhu média si nevyžaduje žiadne zmeny v sieťovej infraštruktúre. Taktiež, aspoň v prípade signalizácie, sa podpora konferenčných hovorov od tých medzi dvoma účastníkmi líši len minimálne.



Obr. 2.2 - Internet Multimedia Conferencing Architecture

2.1 Real-Time Transport Protokol (RTP)

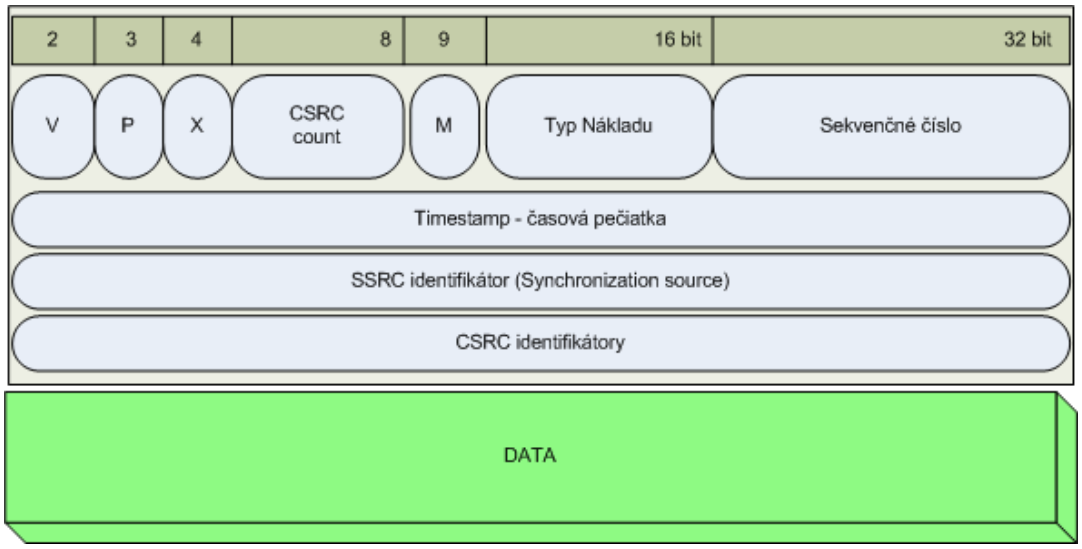
Ako som už uviedol, komunikáciu prostredníctvom VoIP môžeme rozdeliť na dva hlavné celky – signalizáciu, slúžiacu na správu spojenia a na samotný prenos rečových dát v podobe paketov. Práve na tento prenos sa v dnešných VoIP systémoch využíva RTP Protokol (Real-Time Transport Protocol). RTP je transportný protokol, ktorý zabezpečuje doručovanie interaktívnych dát v reálnom čase. Najčastejšie je to video a audio. Protokol poskytuje služby na identifikáciu obsahu (payload), sekvenčné číslovanie, časové pečiatky a monitorovanie doručenia. Typicky RTP využíva UDP (User Datagram Protocol) protokol, môžu byť ale použité aj iné sieťové a transportné protokoly. Aplikácie používajúce RTP sú menej citlivé na stratu paketov ale zase sú veľmi citlivé na oneskorenie (delay), preto je UDP lepšia voľba pre prenos ako TCP (Transmission Control Protocol) u takýchto typov aplikácií.

Treba poznamenať, že RTP sám o sebe neposkytuje žiadny mechanizmus na zaistenie včasného doručenia dát alebo poskytovania nejakej QoS, ale spolieha sa na služby nižších úrovní. Sekvenčné číslo obsiahnuté v RTP umožňuje prijímateľovi rekonštruovať postupnosť paketov od odosielateľa, môže byť ale tiež použité na určenie správnej pozície paketu, napr. pri dekódovaní videa bez nutnosti dekódovať predchádzajúce pakety.

RTP nepoužíva žiadne štandardné TCP alebo UDP porty na ktorých by komunikoval. Jediný štandard ktorého sa drží je, že UDP komunikácia prebieha na párnom porte a ďalší vyšší nepárny port je použitý pre RTP Control Protocol (RTCP) komunikáciu. Aj keď na to neexistuje žiadny štandard, RTP je všeobecne nakonfigurované používať porty od 16384 do 32767.
Podľa RFC 1889 služby poskytované RTP zahŕňajú:

- Identifikácia typu „nákladu“ – Indikácia druhu prenášaného obsahu
- Číslovanie sekvencie – PDU číslo sekvencie
- Časová značka – časová pečiatka obsahu prenášaného v PDU
- Monitorovanie doručenia

Prenos dát je rozšírený o kontrolný protokol (RTCP) ktorému sa budem podrobnejšie venovať neskôr. Tento protokol umožňuje monitorovanie doručenia dát v rozsahu veľkých multicastových sietí a poskytuje obmedzenú funkciu kontroly a identifikácie. RTP aj RTCP sú navrhnuté tak, aby boli nezávislé od podliehajúcej sieťovej a transportnej vrstvy. RTCP poskytuje informácie o prijímanej kvalite ktoré môže daná aplikácia využiť pre lokálne prispôsobenie toku dát. Jeden príklad takéhoto využitia je aplikácia, ktorá v prípade, že nastáva zahltenie siete, sa môže rozhodnúť znížiť bit rate dátového toku.



Obr. 2.1-1 - Štruktúra RTP Protokolu

Popis štruktúry RTP protokolu :

- V – Identifikuje RTP verziu
- P – Padding (výplň). Ak je toto pole nastavené, paket bude na konci obsahovať jeden alebo viacero dodatočných oktetov, ktoré nie sú súčasťou nákladu (payload).
- X – Extension bit. Keď je tento nastavený, je fixná hlavička nasledovaná presne jedným rozšírením hlavičky s definovaným formátom.
- CSRC počet – Obsahuje počet CSRC identifikátorov ktoré nasledujú za fixnou hlavičkou.
- M – Marker. Interpretácia markera alebo návestia je definovaná profilom. Jeho zámerom je označovať dôležité udalosti v rámci vysielania streamu.
- Typ nákladu – Identifikuje formát RTP nákladu a určuje jeho interpretáciu danej aplikácii. Profil špecifikuje implicitné statické mapovanie kódov typu nákladu do jeho formátu. Dodatočné kódy typov nákladov sa môžu dodatočne dynamicky definovať pomocou nie-RTP prostriedkov.
- Sekvenčné číslo – Inkrementuje o jednotku každý odoslaný RTP paket a môže byť použitý prijímačom na detekciu straty paketov a na obnovenie sekvencie paketov.
- Timestamp – Časová pečiatka, odráža vzorkovací okamih prvého oktetu RTP dátového paketu. Táto pečiatka musí byť odvodená od hodín, ktoré sa inkrementujú monotónne a lineárne v čase, aby sa mohla vykonávať synchronizácia a prepočítavanie Jitter (variabilné oneskorenie) chvenia.
- SSRC – Synchronizačný zdroj. Tento identifikátor je vybraný náhodne so zámerom aby žiadne dva synchronizačné zdroje v rámci tej istej RTP relácie nemali ten istý SSRC identifikátor.
- CSRC – Zoznam identifikátorov prispievateľov do RTP streamu.

Samotný náklad čiže payload, jeho veľkosť, závisí od požitého multimediálneho kodeku o ktoré budú popísané v nasledujúcich kapitolách.

2.2 RTP Control Protokol (RTCP)

RTP Control Protocol je založený na periodickom vysielaní kontrolných paketov všetkým účastníkom videokonferencie. Používa pritom rovnaké distribučné mechanizmy ako dátové pakety. Protokol nižšej úrovne musí poskytovať multiplexovanie dátových a kontrolných paketov, napríklad použitím rôznych portov pri UDP protokole. RTCP pritom plní štyri funkcie:

1. Hlavnou funkciou je poskytovanie spätnej odozvy na kvalitu distribuovaných dát. Je to nedeliteľná súčasť úloh RTP ako transportného protokolu. Spätná odozva môže byť priamo použitá k riadeniu adaptívneho kódovania alebo použitím odozvy od prijímateľov na odhaľovanie chýb v distribúcii dát. Funkcia spätnej odozvy je realizovaná prostredníctvom RTCP sender a receiver reports (správy odosielateľa a prijímateľa).
2. RTCP prenáša konštantný identifikátor RTP zdroja nazývaný CNAME (canonical name). Zatiaľ čo SSRC identifikátor sa môže zmeniť v prípade konfliktu alebo reštartovania aplikácie, CNAME potrebuje prijímateľ na udržanie zoznamu účastníkov.
3. Prvé dve funkcie vyžadujú, aby účastníci odosielali RTCP pakety, preto musí byť ich množstvo riadené podľa počtu účastníkov. Tým, že každý účastník odosiela kontrolné pakety všetkým účastníkom, každý si môže uchovávať informáciu o počte účastníkov. Toto číslo je použité k výpočtu tempa, akým sú odosielané kontrolné pakety.
4. Štvrtá, voliteľná funkcia, je poskytovanie minimálnych kontrolných informácií o relácii, ako napr. zobrazovanie identifikácie účastníkov.

Funkcie 1 až 3 by mali byť použité vo všetkých prostrediach, hlavne ale v prostredí IP multicastu.

2.3 Signalizácia

Nadviazanie telefónneho hovoru v hlasových sieťach (VoIP aj klasickej PSTN) sa deje v dvoch fázach. V prvej fáze sa za pomoci signalizačných protokolov dohodne akým spôsobom budú koncové stanice prepojené, aké kodeky pre kompresiu zvuku budú použité a následne potom v druhej fáze vykoná prepojenie koncových staníc pomocou špecifikovaného dátového protokolu.
V PSTN sieťach sa na signalizáciu väčšinou používa protokol SS7. SS7 je robustný signalizačný protokol používaný pre medzinárodnú signalizáciu medzi ústredňami, čo ale znamená že je zložitý a pre koncové zariadenia by bol asi ťažko implementovateľný. Preto sa vo VoIP sieťach vyvinuli nové signalizačné protokoly ako napr. H323, SCCP, SIP, IAX. Ďalej rozoberiem niektoré najznámejšie a najpoužívanejšie signalizačné protokoly.

2.3.1 H.323

H.323 je štandard ITU-T a bol vyvinutý v ”Enterprise LAN community“ ako video konferenčný protokol podobný signalizačným protokolom používaných v ISDN sieťach ako napr. Q.931. Primárne bol určený pre použitie vo vysokorýchlostných LAN sieťach

ktoré priamo neposkytujú QoS. H.323 popisuje sadu protokolov z ktorých každý samostatne popisuje a vykonáva určitú činnosť (autentifikáciu, presmerovanie hovorov, nadväzovanie spojenia, ...)

H323 architektúra pozostáva z nasledovných komponentov:

- Terminál
- Gateway
- Gatekeeper
- MCU

Terminál - je základným a jediným povinným komponentom H.323 siete. Používa sa na obojsmernú komunikáciu v reálnom čase. Môže to byť PC alebo špecializované zariadenie na ktorom beží H.323 stack. Záväzne musí podporovať audio služby, video a dáta sú voliteľné. Je schopný komunikovať s inými multimediálnymi terminálmi v iných sieťach.

Gateway - zabezpečuje spojenie H.323 siete s inou sieťou. Je to vlastne prekladač protokolov medzi rôznymi sieťami.

Gatekeeper - je voliteľným komponentom, ktorý ma na starosti služby ako autorizácia, autentifikácia, preklad telefónnych čísel na IP adresy, smerovanie hovorov atď. Ide vlastne o analógiu s inteligentnou ústredňou.

MCU (Multipoint Control Unit) - zaisťuje komunikáciu troch a viac terminálov, teda konferenčné hovory.

Táto architektúra bola v začiatkoch VoIP veľmi rozšírená ale vďaka jej nadmernej zložitosti a komplexnosti je v poslednej dobe vytlačovaná iným omnoho flexibilnejším protokolom a tým je Session Initiation Protocol – SIP. SIP je protokol čoraz viacej nasadzovaný pre signalizáciu nielen vo VoIP systémoch a v ďalšom texte sa budem zaoberať hlavne aspektmi signalizácie za použitia tohto protokolu.

2.3.2 Session Initiation Protocol (SIP)

SIP (Session Initiation Protocol) pochádza z dielne IETF a bol pôvodne určený aby tvoril mechanizmus pre pozývanie ľudí do veľkokapacitných multipoint konferencií na Internet Multicast Backbone (Mbone). V tomto štádiu internetová telefónia VoIP v podstate ešte neexistovala. Časom sa ale usúdilo, že práve SIP by mohol byť použitý pre vytváranie point-to-point konferencií – telefónnych hovorov.
Prístup aký zvolil SIP je klasický v smere ako sa inovuje Internet: vytvor len to čo je potrebné, adresuj len to čo chýba v existujúcich mechanizmoch. Pretože prístup aký sa zvolil pri SIPe je modulárny a nezávislý od podliehajúcich protokolov alebo architektúry, a pretože protokol sám o sebe je veľmi jednoduchý, SIP sa ujal ako alternatíva k H.323 a ďalším proprietárnym protokolom pre účely signalizácie v sieťach.
Filozofia IETF je jednoduchosť: špecifikuj iba to čo je potrebné aby bolo špecifikované. A SIP sa v tomto v ničom nelíši. Vyvinutý čisto ako mechanizmus pre manažment relácií, nevie o detailoch relácie, iba zahajuje, modifikuje a ruší relácie. Táto jednoduchosť znamená, že je ľahko rozšíriteľný a je použiteľný v rôznych architektúrach a scenároch pre distribúciu multimediálneho obsahu.
SIP je tzv. request- response protokol ktorý sa pri bližšom pohľade podobá na dva iné internetové protokoly, HTTP a SMTP (web a e-mail). V dôsledku toho môže SIP bez problémov koexistovať s existujúcimi internetovými aplikáciami. Takto sa VoIP telefónia stáva ďalšou internetovou aplikáciou a ľahko sa integruje do iných internetových služieb.

SIP je jednoduchá sada nástrojov ktorú môžu poskytovatelia služieb použiť k vytvoreniu konvergovaných hlasových a multimediálnych služieb.
Na to aby sme mohli poskytovať telefónne služby je potreba množstvo ďalších štandardov a protokolov – spomeniem napr. protokol pre zabezpečenie prenosu (RTP), autentifikáciu a autorizáciu užívateľov (RADIUS, DIAMETER), adresárové služby (LDAP), garancia hlasovej kvality (RSVP, YESSIR) a protokoly pre zabezpečenie spolupráce s inými dnešnými telefónnymi sieťami.

2.3.2.1 Kooperácia

SIP bol navrhnutý, aby riešil iba obmedzenú časť problémov a aby spolupracoval so širokým spektrom existujúcich a budúcich protokolov pre IP telefóniu. Momentálne SIP poskytuje štyri základné funkcie. Umožňuje lokalizovanie užívateľov tzv. User Location (preklad z užívateľského mena na korektnú sieťovú adresu). SIP poskytuje funkciu pre dohodu o rozšíreniach tzv. Feature Negotiation, tak aby všetci účastníci danej relácie mohli súhlasiť s rozšíreniami ktoré sa budú medzi nimi používať. SIP je mechanizmus pre správu hovorov - Call Management, napríklad manipulácia s účastníkmi ako pridávanie a odoberanie z relácie. A nakoniec SIP dokáže meniť rozšírenia a parametre danej relácie pokým ešte stále trvá. Všetky ostatné funkcie sú zabezpečované inými protokolmi.
To vedie k tomu čo SIP nezabezpečuje. SIP nie je protokol pre popis relácie (Session Description Protokol) a nerobí kontrolu konferenčných hovorov. SIP nie je protokol pre rezerváciu zdrojov (resource reservation protokol) a nezabezpečuje žiadne funkcie QoS. SIP ale dokáže pracovať v množine protokolov ktoré tieto funkcie dokážu zabezpečiť ale SIP samotných ich nevykonáva. SIP spolupracuje so SOAP, HTTP, XML, VXML, WSDL, UDDI, SDP a množstvom iných protokolov. Každý plní svoju špecifickú funkciu.

Nie je pochýb o tom, že SIP bol navrhnutý ako modulárny komponent väčšieho riešenia IP telefónie a preto dobre spolupracuje s množstvom protokolov založených na IP. Ale SIP je dokonca ešte "priateľskejší" pretože spolupracuje aj s protokolmi ktorých funkcie sa prelínajú s tými jeho. V blízkej dobe bude SIP musieť koexistovať s protokolmi ktorých funkcie sa prelínajú ako H.323, MGCP a MEGACO.

H.323 siete sú už rozšírené v mnohých častiach sveta a sieťoví operátori majú záujem v rozrastajúcich sa možnostiach SIP sietí. Produkty pre konverziu z H.323 do SIP sietí už sú na trhu. MGCP a MEGACO môžu tiež len získať v spolupráci so SIP keďže sami o sebe nedokážu vytvoriť kompletný IP telefónny systém. Tieto protokoly sa nachádzajú architektonicky pod SIP protokolom a môžu získať na funkcionalite keď budú kontrolované pomocou SIP.

Jednoducho napísané, SIP je dôležitý protokol ktorý sa čím ďalej tým viac rozširuje. SIP je katalytický protokol ktorý poskytuje kľúčovú signalizačnú zložku, ktorá dokáže zmeniť sieť pre IP telefóniu na skutočnú IP telekomunikačnú sieť - sieť schopnú poskytnúť konvergované služby ďalšej generácie. SIP je jednoduchý a pritom veľmi užitočný protokol, ktorý popri tom veľmi dobre spolupracuje s inými protokolmi.

Ďalej sa budem snažiť podrobnejšie ozrejmiť fungovanie tohto protokolu keďže kvalita celkového hovoru od jeho naviazania až po jeho zrušenie závisí z jednej časti práve od signalizačného protokolu. V rámci mojich meraní taktiež používam práve VoIP infraštruktúru založenú na SIP protokole.

2.3.2.2 Adresácia

SIP obsahuje elementy dvoch, veľmi rozšírených internetových protokolov. Hyper Text Transport Protokol (HTTP) používaný pre prehliadanie webu a Simple mail transport protokol (SMTP) používaný pre e-mailové služby. Z HTTP si SIP vzal klient-server dizajn a používanie URL a URI adries. Uniform Resource Locators - URL, sú mená reprezentujúce adresy alebo miesta v sieti internet. URL sú navrhnuté tak, aby zahrňovali

široké spektrum protokolov a typov zdrojov internetu. Základná forma URL je schéma: špecifikátor, napríklad *http://www.kis.fri.uniza.sk/sip.html*. Značenie http v tomto prípade značí schému alebo protokol ktorý sa má použiť, v tomto príklade HTTP. Za špecifikátorom nasleduje doménové meno (*kis.fri.uniza.sk*) ktoré sa dá preložiť na IP adresu a meno súboru (*sip.html*). URL tiež zvyknú obsahovať iné parametre alebo kvalifikátory vzťahujúce sa na prenos. Napríklad *ftp://kis.fri.uniza.sk* špecifikuje FTP prenos čiže aj zodpovedajúci protokol. Nové schémy pre URL a nové protokoly sa dajú jednoducho začleniť. Množstvo protokolov sa odkazuje na URL ale SIP sa odkazuje na URI (Uniform Resource Identifier). Je to hlavne kvôli aspektom mobility SIP protokolu čo znamená, že daná adresa nie je viazaná na špecifické fyzické zariadenie ale miesto toho je to len logická entita, ktorá sa môže pohybovať a meniť svoju polohu hocikde na sieti.
Schémy "sip:" alebo "sips:" sú detailne popísané v RFC 2369. Využívajú schému podobnú schéme mailto, podporujúcej špecifikovanie polí hlavičky REQUEST a tela správy SIP. Toto umožňuje špecifikovať polia subject, typ média alebo prioritu relácie započatej použitím URI na web stránke alebo v e-mailovej správe. Vo všeobecnosti v prípade SIP URI je forma nasledujúca:

sip:user:password@host:port;uri-parametre?hlavičky

2.3.2.3 Architektúra a SIP entity

V rámci SIP sa nachádzajú dve základné entity: SIP User Agent (SIP UA) a SIP network server. User agent je koncový komponent systému pre hovor a SIP server je sieťové zariadenie, ktoré spravuje signalizáciu týchto hovorov. Sám User agent má v sebe klientskú časť tzv. User Agent Client a serverovskú časť tzv. User Agent Server. Klientská časť inicializuje hovory a serverovská časť na ne odpovedá. Toto umožňuje peer-to-peer hovory s použitím klient-server protokolu.
SIP User Agenti môžu byť odľahčení klienti vhodní pre koncové zariadenia ako prenosné headsety alebo PDA. Alternatívne to môžu byť desktopové aplikácie v spojení s iným softvérom ako napríklad správcovia kontaktov. Hlavnou funkciou SIP serverov je poskytovať preklad adries a lokalizácia užívateľov, pretože je nepravdepodobné že by volajúci poznal IP adresu toho komu volá, a preposielať správy ďalším serverom použitím tzv. next hop routing protocols.
SIP server môže operovať v dvoch odlišných módoch: stavovom a bezstavovom. Rozdiel v týchto dvoch módoch je, že serve pracujúci v stavovom režime si zapamätá prichádzajúcu požiadavku, spolu s odpoveďami ktoré pošle späť a ďalšími požiadavkami ktoré rozpošle. V skratke stavový server si je vedomý prebiehajúcej relácie. Na druhú stranu bezstavový server všetky informácie ktoré spracuje následne zabudne. Tieto bezstavové servery sa spravidla nachádzajú na chrbtici SIP infraštruktúry a stavové servery zase bližšie koncovým zariadeniam kde kontrolujú domény užívateľov.

UAC – User Agent Client - je jedným z dvojky klientských komponentov, druhý je UAS ako user agent server. UAC je aplikácia ktorá môže inicializovať šesť základných SIP požiadaviek (metód) na UAS : INVITE, ACK, OPTIONS, BYE, CANCEL a REGISTER.
Keď sa SIP relácia inicializuje UAC SIP komponentom, UAC určí informácie ktoré sú potrebné pre požiadavku, a to protokol, port a IP adresu UAS ktorému sa daná požiadavka má zaslať. Daná správa sa potom s danou metódou zašle UAS ktorý ju spracuje.

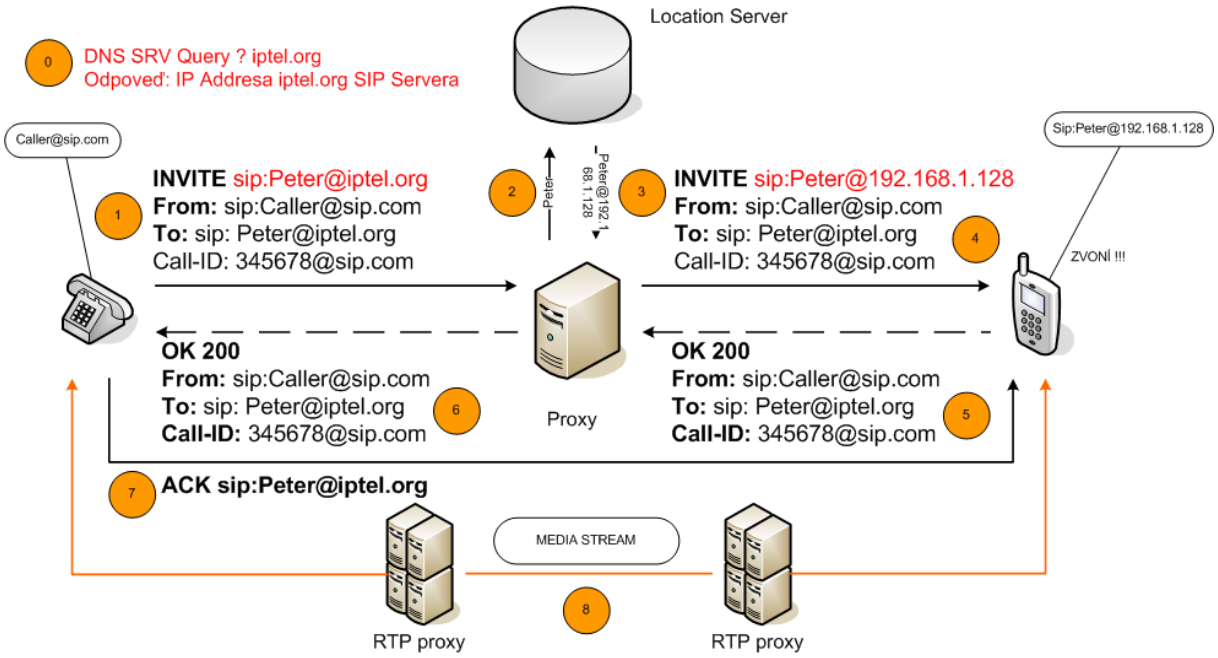
UAS – User Agent Server - je server na ktorom beží aplikácia zodpovedná za prijímanie SIP požiadaviek od UAC, a posiela späť odpovede na tieto požiadavky späť UAC. UAS môže poslať UAC späť viacero odpovedí nie nutne len jednu odpoveď. Komunikácia medzi UAC a UAS je klient-server a peer-to-peer.

Back-to-Back User Agent - Takzvaný back-to-back User Agent (B2BUA) je typ SIP zariadenia ktoré prijíma SIP požiadavky, potom preformuluje požiadavku a pošle do siete ako novú požiadavku. Odpovede na tieto požiadavky sú tiež reformulované a poslané späť v opačnom smere. Napríklad B2BUA môže byť použitý ako implementácia služby anonymizér v ktorej dvaja SIP UA môžu komunikovať bez toho aby sa hociktorý z účastníkov dozvedel URI, IP adresu alebo akúkoľvek inú informáciu. Aby sa toho dosiahlo, musí B2BUA preformulovať požiadavky s úplne novými poľami FROM, VIA, CONTACT, CALL-ID a SDP informácie o médiu, tiež musí odstrániť akékoľvek iné pole SIP hlavičky, ktoré by mohlo obsahovať nejaké informácie o volajúcom. Vrátená odpoveď by mala tiež zmenené polia CONTACT a SDP informácie o médiu. Modifikované SDP by smerovalo na B2BUA samotný., ktorý by preposielal RTP pakety od volaného volajúcemu a naopak. Týmto spôsobom sa nik z koncových užívateľov nedozvie akúkoľvek identifikujúcu informáciu o druhom volajúcom počas nadväzovania spojenia. (samozrejme že volajúci musí poznať URI volaného aby sa hovor mohol uskutočniť).
Niekedy sú B2BUA nasadené na poskytovanie iných SIP služieb. Avšak porušujú end-to-end náturu internetového protokolu ako je SIP. B2BUA je tiež stavový SPF (single point of failure) v sieti, čo znamená, že ich používaním sa znižuje spoľahlivosť SIP relácií cez internet. Takto upravené relácie trpia zvýšenou latenciou a väčšou stratou paketov, čo v konečnom dôsledku znižuje kvalitu hovoru. Geografické rozmiestnenie B2BUA môže tieto efekty zmierniť, ale problém zvolenia toho najlepšieho B2BUA pre tú ktorú reláciu je veľmi ťažký, keďže zdrojová ani cieľová IP adresa média nie je známa až pokým sa daná relácia nenadviaže (s 200 OK správou). Najčastejšou formou B2BUA nachádzajúcich sa v SIP sieťach sú ALG (Application layer gateways). Niektoré firewally majú ALG funkcionalitu zabudovanú, čo umožňuje firewallu povoliť SIP a RTP traffic a pritom stále udržovať vysoký stupeň ochrany.

Proxy server - SIP proxy server je zariadenie ktoré obdrží SIP požiadavku od UA alebo iného proxy servera a koná v záujme UA tak, že preposiela alebo odpovedá na tieto požiadavky. Proxy server nie B2BUA pretože môže modifikovať požiadavky a odpovede iba v obmedzenej miere špecifikovanej v RFC 3261. Tieto pravidlá sa snažia zachovávať end-to-end transparenciu SIP signalizácie a pri tom stále povoľovať proxy serveru ponúkať cenné služby koncovým užívateľom alebo UA. Proxy server máva typicky prístup do databázy užívateľov alebo k nejakej lokačnej službe aby mohol pomáhať v spracovaní požiadavky (určenie ďalšieho hopu). Rozhranie medzi proxy a lokačnou službou nemusí byť uskutočňované pomocou SIP protokolu. Proxy môže použiť akýkoľvek počet a typov databáz v procese spracovania požiadavky. Databázy môžu obsahovať SIP registrácie, informácie o prítomnosti prípadne akékoľvek iné informácie o tom kde sa používateľ nachádza. Proxy server nemusí chápať SIP požiadavku v správe aby ju mohol preposlať - akýkoľvek iný typ požiadavky používa tzv. non-INVITE transakčný model. Proxy by nemalo meniť poradie položiek hlavičiek alebo vo všeobecnosti modifikovať alebo meniť polia hlavičky. Proxy server sa líši od UA alebo brány v troch kľúčových bodoch.

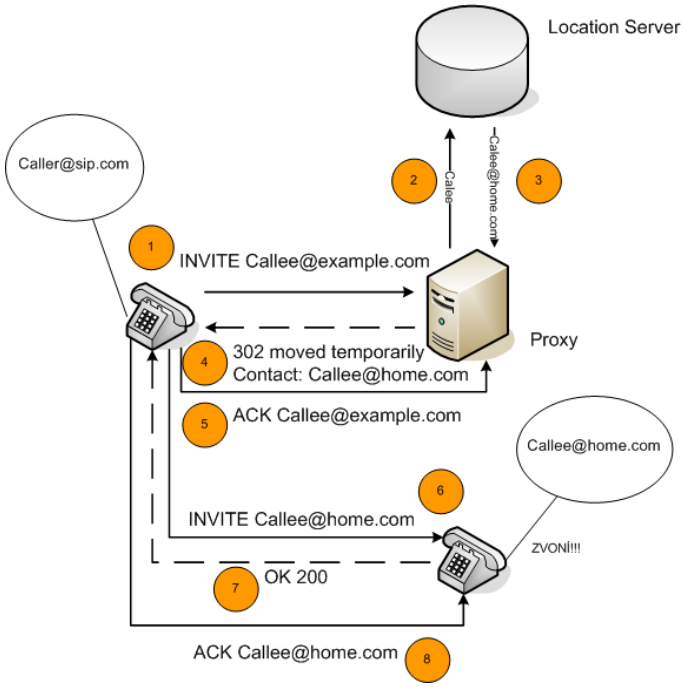
1. Proxy server nevytvára nové požiadavky, iba odpovedá na požiadavky od UA
   (požiadavka CANCEL je v tomto výnimkou)
2. Proxy server nemá žiadne multimediálne funkcie.
3. Proxy server neparsuje telo správ, závisí výlučne na hlavičke SIP správy.

Proxy server môže byť buď stavový alebo bezstavový. Bezstavové proxy spracúva každú SIP požiadavku alebo odpoveď na základe obsahu správy. Ako náhle sa správa raz rozparsuje, spracuje a prepošle ďalej alebo pošle odpoveď, žiadna informácia o tejto správe sa nikde neuloží - nie je uložená žiadna informácia o dialógu. Bezstavové proxy nikdy neposiela opakovanú správu a nepoužíva žiadne SIP časovače. Bezstavová detekcia slučiek použitím VIA poľa hlavičky popísaná v RFC 2543 bola odstránená v RFC 3261 v prospech používania nepovinného poľa hlavičky Max-Forwards vo všetkých požiadavkách. Najbežnejší typ SIP proxy servera je transakčno stavové proxy. Stavové proxy si udržuje stav transakcie ale iba počas trvania požiadaviek.



Obr. 2.3.2.3 – 1 – Schéma zobrazujúca funkcionalitu Proxy servera v SIP infraštruktúre

Redirect Server - Redirect alebo presmerovávací server je typ servera ktorý posiela odpovede ale nepreposiela ďalej požiadavky. Podobne ako proxy server používa aj redirect server databázu alebo lokačnú službu na vyhľadanie používateľa. Informácia o umiestnení sa avšak pošle späť volajúcemu v redirect odpovedi typu 3XX, a po obdržaní ACK sa daná transakcia ukončí.

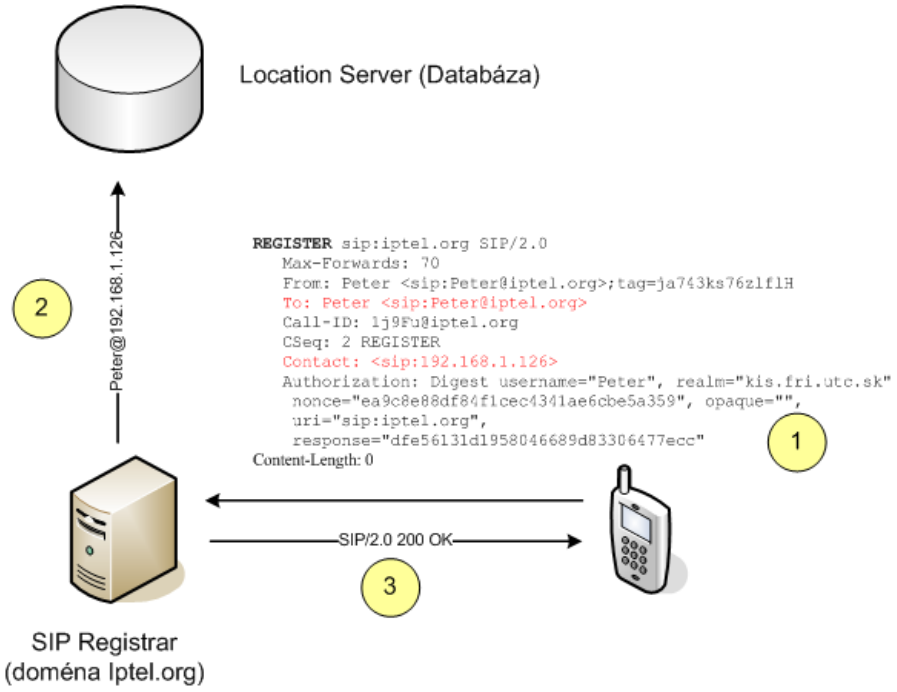


2.3.2.3 – 2 – Schéma zobrazujúca funkcionalitu Redirect servera

Location Server - Volajúci sa môže postupom času pohybovať medzi niekoľkými koncovými zariadeniami. Tieto polohy môžu byť dynamicky registrované so SIP serverom. Keď je server dotazovaný na polohu volaného, vráti zoznam možných polôh. Lokačný server v SIP systéme v podstate vygeneruje zoznam a posunie ho SIP serveru. To čo sa

vykoná s týmto zoznamom záleží od typu servera ktorý si ho vyžiadal. SIP redirect server vráti zoznam klientovi no SIP proxy server skúša jednotlivé adresy buď sekvenčne alebo paralelne až pokým je hovor úspešný alebo volaný zruší hovor.

Registrar **-** Registračný server, tiež známy pod pojmom Registrar, prijíma požiadavky typu REGISTER, všetky ostatné požiadavky server odmietne odpoveďou 501 Not implemented. Kontaktná informácia z požiadavky je potom poskytnutá ďalším SIP serverom v tej istej administratívnej doméne, ako sú proxy alebo redirect servre. V registračnej požiadavke samotnej ,"To" pole hlavičky obsahuje meno zdroja ktorý sa registruje a pole "Contact" zase obsahuje alternatívne adresy alebo aliasy. Registračný server vytvorí dočasné spojenie medzi tzv. Address of Record (AOR) URI v "To" poli a URI zariadenia v poli "Contact". Registračný server zvyčajne vyžaduje od registrovaného užívateľa autentifikáciu, tak aby nebolo možné prichádzajúce hovory odpočúvať neautorizovaným užívateľom. Toto by sa mohlo stať ak by si neautorizovaný používateľ chcel registrovať cudziu URI na svoj telefón. Prichádzajúce hovory by potom boli presmerované na iné fyzické zariadenie. V závislosti na hlavičke Register požiadavky, môže UA získať zoznam momentálnych registrácií, vymazať všetky registrácie alebo pridať registračnú URI do zoznamu.



2.3.2.3 – 3 – Schéma zobrazujúca funkcionalitu Registraru

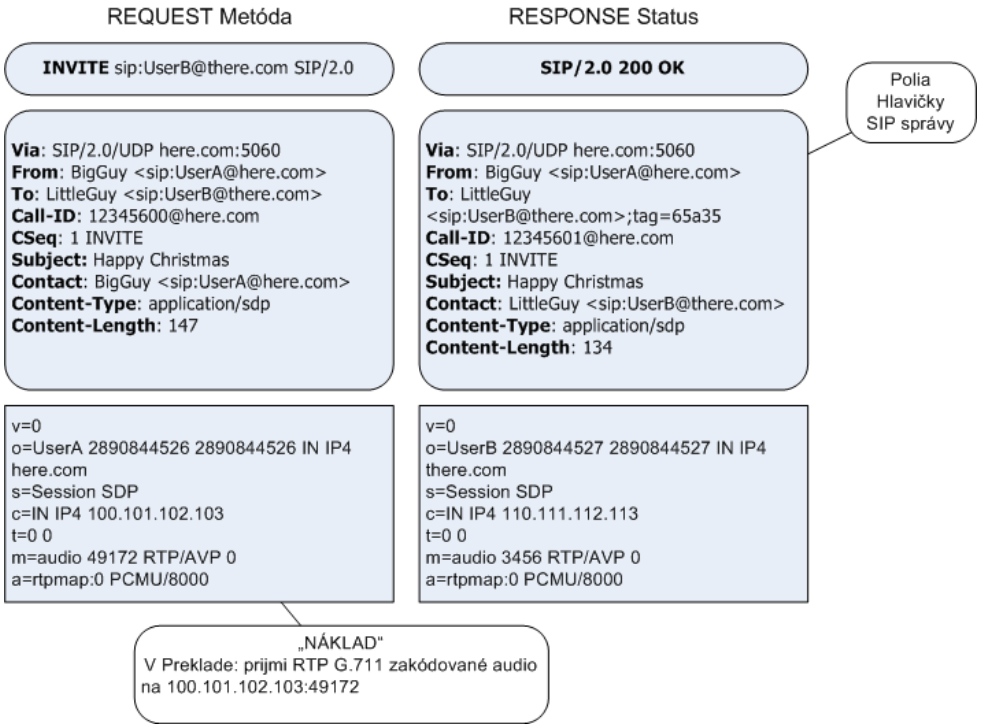
2.3.2.4 SIP správy

SIP je textovo orientovaný protokol a používa UTF-8 znakovú sadu (RFC 2279). SIP správa je buď požiadavka - Request od klienta na server alebo odpoveď - Response od servera klientovi. Oba typy, aj požiadavky aj odpovede používajú základný formát UTF-8 aj keď sa syntax líši v znakovej sade a špecifikách syntaxe. (SIP dovoľuje polia hlavičky, ktoré nie sú platné UTF-8 polia hlavičky). Obidva typy správ pozostávajú zo štartovacieho riadku - start-line, jedného alebo viacerých polí hlavičky, prázdneho riadku indikujúceho koniec polí hlavičky a voliteľné telo správy - message body.

generic-message = start-line
**message-header*
CLRF
(Message Body)

start-line = Request-line / Status-line

Štartovací riadok, každý riadok hlavičky a prázdny riadok musia byť ukončené terminátorom - carriage-return line-feed sequence (CRLF). Prázdny riadok musí v správe byť aj keby správa nemala telo správy. Až na rozdiel v znakových sadách, má väčšina SIP správ podobnú syntax ako HTTP/1.1



Obr. 2.3.2.4 – 1 – Porovnanie štruktúry request a response správ

SIP Request správy

Požiadavky sa obvykle používajú na inicializovanie niektorých akcií alebo na informovanie príjemcu o nejakej požiadavke. Odpovede sú používané na potvrdenie, že požiadavka bola prijatá a spracovaná a obsahujú stav spracovania. SIP požiadavky sú rozlíšené tým, že majú Request-line ako Start-line. Request-line obsahuje názov metódy, Request URI a verziu protokolu oddelené tzv. single space character (SP), prázdnym znakom. Request-line končí CRLF.

Request-line = Metóda SP Request-URI SP SIP Verzia CRLF

Špecifikácia definuje šesť metód: REGISTER - pre registráciu kontaktných informácií, INVITE, ACK a CANCEL pre zostavenie relácie, BYE pre ukončenie relácie a OPTIONS na dotazovanie serverov na ich schopnosti. Rozšírenia SIP môžu definovať dodatočné metódy.
Typická SIP požiadavka môže vyzerať nasledovne:

```
INVITE sip:7170@iptel.org SIP/2.0
Via: SIP/2.0/UDP 195.37.77.100:5040;rport
Max-Forwards: 10
From: "jiri" <sip:jiri@iptel.org>;tag=76ff7a07-c091-4192-84a0-
d56e91fe104f
To: <sip:jiri@bat.iptel.org>
Call-ID: d10815e0-bf17-4afa-8412-d9130a793d96@213.20.128.35
CSeq: 2 INVITE
Contact: <sip:213.20.128.35:9315>
User-Agent: Windows RTC/1.0
Proxy-Authorization: Digest username="jiri", realm="iptel.org",
 algorithm="MD5", uri="sip:jiri@bat.iptel.org",
 nonce="3cef753900000001771328f5ae1b8b7f0d742da1feb5753c",
 response="53fe98db10e1074
 b03b3e06438bda70f"
Content-Type: application/sdp
Content-Length: 451

v=0
o=jku2 0 0 IN IP4 213.20.128.35
s=session
c=IN IP4 213.20.128.35
b=CT:1000
t=0 0
m=audio 54742 RTP/AVP 97 111 112 6 0 8 4 5 3 101
a=rtpmap:97 red/8000
a=rtpmap:111 SIREN/16000
a=fmtp:111 bitrate=16000
a=rtpmap:112 G7221/16000
a=fmtp:112 bitrate=24000
a=rtpmap:6 DVI4/16000
a=rtpmap:0 PCMU/8000
a=rtpmap:4 G723/8000
a=rtpmap: 3 GSM/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
```

Prvý riadok nám vraví, že toto je INVITE správa ktorá sa používa na zostavovanie spojenia. URI prvého riadku - sip:7170@iptel.org sa nazýva Request-URI a obsahuje URI ďalšieho hopu správy. V tomto prípade to bude zariadenie 195.37.77.100 a port 5060. From a To polia hlavičky identifikujú volajúceho a volaného účastníka. Pole hlavičky From obsahuje parameter Tag, ktorý slúži na identifikáciu dialógu. Pole Call-ID je identifikátor dialógu a jeho úlohou je identifikovať správy, ktoré patria k tomu istému hovoru. Takéto správy majú túto hodnotu rovnakú. CSeq sa používa na správu poradia požiadaviek. Pretože môžu byť požiadavky poslané cez nespoľahlivý transportný systém ktorý môže správy poprehadzovať, musí byť v správe prítomné sekvenčné číslo, tak aby príjemca mohol identifikovať retransmisie a požiadavky mimo poradia. Pole hlavičky Contact obsahuje IP adresu a port na ktorom odosielateľ očakáva neskoršie požiadavky od volaného. Hlavička správy je od tela správy oddelená prázdnym riadkom. Telo správy INVITE obsahuje popis typu média akceptovaného odosielateľom a zapúzdrenom v SDP.

Popísal som ako vyzerá požiadavka INVITE ktorá sa používa na zostavenie relácie. Ostatné dôležité požiadavky sú :

ACK - táto správa potvrdzuje volajúcemu príjem požiadavky INVITE. Vytvorenie relácie vyžaduje troj smernú výmenu (3-way hand shake) kvôli asymetrickej povahy pozvánky. Môže to chvíľu trvať než volaný prijme alebo odmietne hovor, takže volaný UA periodicky posiela pozitívne finálne odpovede pokým nedostane ACK (čo indikuje, že volajúci je stále tam a pripravený komunikovať).

BYE - tento typ správy sa používa na zrušenie multimediálnej relácie. Účastník želajúci ukončiť spojenie pošle druhej strane BYE.

CANCEL - tento typ sa používa na zrušenie ešte nie úplne nadviazaného spojenia. Používa sa ak volaný ešte neodpovedal finálnou odpoveďou ale volajúci chce ukončiť volanie. (Typicky keď volaný neodpovedá po určitý čas)

REGISTER - účelom tejto požiadavky je oboznámiť registrar o momentálnom umiestnení užívateľa. Informácie o momentálnej IP adrese a porte na ktorom je užívateľ zastihnuteľný sú posielané v tejto správe. Registrar túto informáciu vyparsuje a vloží do lokačnej databázy. Databáza môže byť neskôr použitá nejakým proxy serverom na smerovanie hovorov k užívateľovi. Registrácie sú časovo obmedzené a musia byť periodicky obnovované. Spomenuté požiadavky nemajú zvyčajne žiadne telo správy keďže žiadne nepotrebujú v daných situáciách (ale môžu mať).

SIP Response správy

Keď nejaký UA alebo proxy server obdrží požiadavku, pošle späť odpoveď. Na každú požiadavku okrem ACK musí byť poslaná odpoveď. (ACK nevyvoláva žiadne odpovede). SIP odpovede sú rozlíšiteľné od požiadaviek podľa Status-line ako Start-line. Status-line pozostáva z verzie protokolu nasledovanou numerickým status kódom a jeho priradená textová podoba, a každý element je rozdelený pomocou SP.

Status-line = SIP-Version SP Status-code SP Reason-phrase CRLF

Typická odpoveď má nasledovnú podobu:

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 192.168.1.30:5060;received=66.87.48.68
From: sip:sip2@iptel.org
To: sip:sip2@iptel.org;tag=794fe65c16edfdf45da4fc39a5d2867c.b713
Call-ID: 2443936363@192.168.1.30
CSeq: 63629 REGISTER
Contact: Msip:sip2@66.87.48.68:5060;transport=udp>;q=0.00;expires=120
Server: Sip EXpress router (0.8.11pre21xrc (i386/linux))
Content-Length: 0
Warning: 392 195.37.77.101:5060 "Noisy feedback tells:
 pid=5110 req_src_ip=66.87.48.68 req_src_port=5060 in_uri=sip:iptel.org
 out_uri=sip:iptel.org via_cnt==1"
```

Ako môžeme vidieť odpovede sú si veľmi podobné s požiadavkami, až na prvý riadok. Reply-code alebo Status kód je trojmiestne číslo od 100 do 699 a indikuje typ odpovede. Existuje šesť typov odpovedí.

1XX - sú dočasné/informačné odpovede. Dočasná odpoveď je odpoveď ktorá vraví svojmu príjemcovi, že prislúchajúca požiadavka bola prijatá ale výsledok spracovania zatiaľ nie je známy. Dočasné správy sú posielané iba v prípade, že spracovanie nekončí okamžite. Vysielajúci UA musí ukončiť retransmisiu požiadaviek v prípade obdržania dočasnej odpovede. Proxy servery posielajú odpovede s kódom 100 keď začínajú spracovávať INVITE správu a UA posielajú odpovede s kódom 180 (Ringing) čo znamená že telefón volaného zvoní.

2XX - sú pozitívne ukončovacie odpovede. Ukončovacia odpoveď je najvyššia odpoveď ktorú pôvodca požiadavky kedy dostane. Preto ukončovacie odpovede vyjadrujú výsledky spracovania prislúchajúcej požiadavky. Ukončovacie odpovede tiež ukončujú transakcie. Odpovede s kódom od 200 do 299 sú pozitívne odpovede, čo znamená, že požiadavka

bola spracovaná úspešne a prijatá. Napríklad kód 200 OK odpoveď sa posiela keď používateľ akceptuje pozvánku na reláciu (INVITE). UAC môže obdržať niekoľko 200 odpovedí na jedinú požiadavku INVITE. To je zásluhou forking proxy servra, ktorý naklonuje požiadavky tak že budú zaslané niekoľko UAS a každý z nich akceptuje pozvánku. V tomto prípade ja každá odpoveď rozlíšená parametrom Tag v "To" poli hlavičky. Každá odpoveď reprezentuje odlišný dialóg s jednoznačným identifikátora dialógu.

3XX - odpovede sú používané na presmerovanie volajúceho. Presmerovacia odpoveď podáva informácie o používateľovom novom umiestnení alebo o alternatívnej službe ktorá môže zabezpečiť úspešný hovor. Presmerovacie odpovede sú zvyčajne zasielané proxy servermi. Keď proxy obdrží požiadavku a nechce alebo nemôže ju z nejakého dôvodu spracovať, pošle volajúcemu presmerovaciu odpoveď a vloží do nej iné umiestnenie ktoré môže volajúci vyskúšať. Môže to byť adresa iného proxy servera alebo momentálna poloha volaného. (z lokačnej databázy vytvorenej registrarom). Volajúci by potom mal znova preposlať svoju požiadavku na nové umiestnenie. 3XX správy sú konečné.

4XX - sú nagatívne konečné odpovede. 4xx odpoveď znamená, že problém je na strane volajúceho. Požiadavka nemôže byť spracovaná pretože obsahuje zlú syntax alebo nemôže byť spracovaná na danom serveri.

5XX - znamená že problém je na strane servera. Požiadavka je očividne správna ale server zlyhal v jej spracovaní. Klienti by mali zvyčajne preposlať požiadavku neskôr.

6XX - tento kód znamená že požiadavka nemôže byť spracovaná žiadnym zo serverov. Táto odpoveď je obvykle poslaná serverom ktorý má kompletné informácie o danom používateľovi. UA zvyčajne posielajú odpoveď 603 Decline Response keď používateľ nechce participovať v relácii.

Prvý riadok správy obsahuje okrem kódu aj frázu. Kódové označenie je zamýšľané byť čítané strojmi. Nie je veľmi užívateľsky prívetivé ale je jednoducho parsovateľné a pochopené strojmi. Táto fráza zvyčajne obsahuje čitateľnú správu popisujúcu výsledok spracovania. UA by mal túto frázu zobraziť užívateľovi.
Požiadavka, ku ktorej prislúcha korešpondujúca odpoveď je identifikovaná použitím poľa hlavičky CSeq. Mimo sekvenčného čísla obsahuje toto pole hlavičky metódu korešpondujúcej požiadavky. V našom prípade to bola REGISTER požiadavka.

2.4 Audiokodeky

Kodek (Coder/Decoder) konvertuje analógový signál na digitálny prúd dát a ďalší identický kodek na vzdialenom konci komunikačného kanálu tento prúd dát prekonvertuje späť na analógový signál. Vo svete VoIP nám kodeky slúžia na kódovanie reči v rámci paketových sietí akou je aj IP sieť. Kodeky používané pre VoIP sa tiež nazývajú „vocoders“ ako „voice-encoders“. Kodeky vo všeobecnosti poskytujú schopnosť kompresie pre zníženie nárokov na sieťovú šírku pásma. Jednou zo základných vlastností paketov používaných pre prenos hlasu v počítačových sieťach je schopnosť potlačiť ticho v rozhovore. Je to zabezpečené tak, že hlavička RTP obsahuje sekvenčné číslo a časovú pečiatku, ktoré umožňujú príjemcovi rozoznať medzi stratou paketov a časom, keď nie sú prenášané dáta a vysielateľ je ticho. Táto vlastnosť nesúvislého prenosu, čiže potlačenia ticha, môže byť použitá s ktorýmkoľvek typom prenášaných dát a s ktorýmkoľvek druhom použitého audio kódovania. Samozrejme, aj príjemca dát musí byť na túto vlastnosť pripravený.
RTP časovač používaný pri generovaní časovej pečiatky je nezávislý na počte kanálov a druhu kódovania. Pre N kanálové kódovanie, každá vzorkovacia perióda generuje N vzoriek. Vzorky všetkých kanálov patriace k jednému vzorkovaniu musia byť obsiahnuté v

tom istom pakete. Pre rôzne druhy kódovania musia byť vzorkovacie frekvencie vybraté z týchto : 8000, 11025, 16000, 22025, 24000, 32000, 44100 alebo 48000 Hz. Poznáme dva druhy kódovania:

- Kódovanie založené na vzorkách (sample-based) - V tomto kódovaní je každá audio vzorka reprezentovaná pevne stanoveným počtom bitov. RTP audio paket môže obsahovať hocijaké množstvo zvukových vzoriek, ktorým je určené trvanie audio paketu. Ak sa pri tomto kódovaní vytvára jeden alebo viac oktetov na vzorku a zároveň sú v tom istom čase odoberané vzorky v druhom kanáli, potom sa tieto oktety môžu spájať do takzvaných postupných oktetov. Napríklad pri dvojkanálovom kódovaní ide najprv ľavý kanál, prvá vzorka, pravý kanál prvá vzorka, potom ľavý kanál druhá vzorka, pravý kanál druhá vzorka atď.

- Kódovanie založené na rámcoch (frame-based) - Toto kódovanie zakóduje audio blok pevne stanovenej dĺžky do ďalšieho bloku komprimovaných dát, väčšinou tiež pevne stanovenej dĺžky. Pre kódovanie založené na rámcoch sa vysielateľ môže rozhodnúť skombinovať viaceré takéto rámce do jedného RTP paketu.

Najčastejšie používané audiokodeky pre VoIP sú zobrazené v tabuľke.

Kodek	**Algoritmus**	**Bit Rate (Kb/s)**	**Ethernet Bit-Rate (Kb/s)**
ITU G.711	PCM (Pulse Code Modulation)	64	87,2
ITU G.722	SBADPCM (Sub-Band Adaptive Differential Pulse Code Modulation)	48, 56 and 64	
ITU G.723	Multi-rate Coder	5.3 and 6.4	21,9 a 20,8
ITU G.726	ADPCM (Adaptive Differential Pulse Code Modulation)	24, 32	47,2 alebo 55,2
ITU G.727	Variable-Rate ADPCM	16-40	
ITU G.728	LD-CELP (Low-Delay Code Excited Linear Prediction)	16	31,5
ITU G.729	CS-ACELP (Conjugate Structure Algebraic-Code Excited Linear Prediction)	8	31,2
ILBC	Internet Low Bitrate Codec	13.33 a15.20	
Speex	CELP (Code Excited Linear Prediction)	2.15-44.2	
GSM - Full Rate	RPE-LTP (Regular Pulse Excitation Long-Term Prediction)	13	
GSM - Enhanced Full Rate	ACELP (Algebraic Code Excited Linear Prediction)	12.2	
GSM - Half Rate	CELP-VSELP (Code Excited Linear Prediction - Vector Sum Excited Linear Prediction)	11.4	
DoD FS-1016	CELP (Code Excited Linear Prediction)	4.8	

Tab. 2.4 – 1 Audio Kodeky a ich bitrate hodnoty

Výber toho ktorého kodeku závisí hlavne na koncovom zariadení a na subjektívnej kvalite hlasu ktorú kodek poskytuje. Najlepšiu kvalitu reči poskytuje kodek G. 711 (Podľa hodnoty MOS ktorá bude ozrejmená ďalej v texte) avšak má aj najväčšie požiadavky na šírku pásma. Len pre príklad keby sme mali štyroch súčasne volajúcich účastníkov a rečový kodek by bol G.711 tak nároky na šírku pásma by boli 4 * 87,2 Kb/s = 349 Kb/s v jednom smere čiže treba počítať aj s prichádzajúcimi dátami v rovnakom objeme.

2.5 Meranie kvality hlasu - MOS

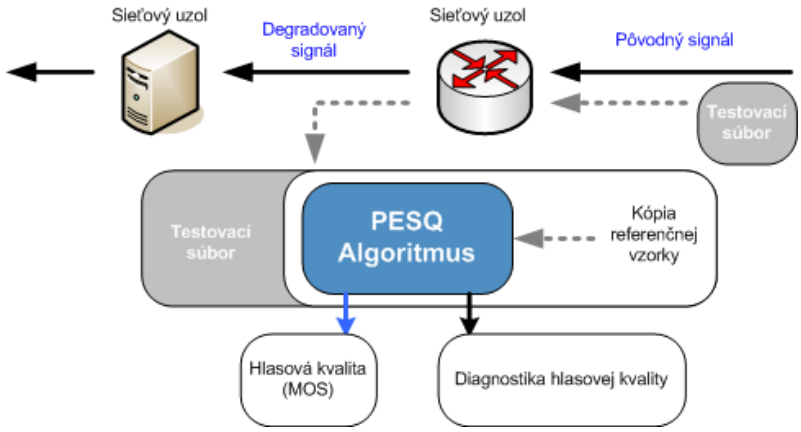
V tejto práci sa odvolávam na meranie kvality daného hovoru vyjadrenou hodnotou MOS. Mean Opinion Score (MOS) sa bežne používa pre ohodnotenie kvalitu telefónneho rozhovoru vyjadrenú na stupnici od 1 do 5, kde 5 je najlepšia kvalita. Hodnotenie číslom 4 sa vo všeobecnosti považuje za kvalitu známu z dnešných PSTN/TDM sietí. MOS je funkcia mnohých faktorov, zahrňujúcich typ siete, použitý kodek, kabeláž a vybavenie a dokonca aj vstupné zariadenia, ktoré sa používajú (headsety).

MOS bol pôvodne určený na subjektívne sluchové testovanie, kde sa skupina trénovaných expertov pokúšala hlasovej vzorke priradiť nejakú priemernú hodnotu. Testovacie vybavenie dnes vypočítava MOS použitím sofistikovaných algoritmov, ktoré sú navrhnuté aby veľmi presne aproximovali výsledky subjektívnych sluchových testov. MOS je celkové ohodnotenie hlasovej kvality, zahrňujúc tucty faktorov do jedného výsledného skóre. Pretože je to ale všeobecná mierka odzrkadľujúca mnoho faktorov, nemalo by sa MOS používať ako výhradné hodnotenie hlasovej kvality. Niektoré faktory ako echo, oneskorenie alebo sila hovoru by pri istých algoritmoch MOS skóre výrazne neznížili avšak subjektívny pocit z hovoru by sa zhoršil výrazne. Najlepšie hodnotenie kvality preto pozostáva z monitorovania jednotlivých faktorov hlasu ako je šum, oneskorenie, jitter a strata paketov spolu s hodnotením MOS pri použití viacerých algoritmov.

Existuje niekoľko štandardizovaných MOS algoritmov, každý pôvodne navrhnutý pre konkrétne použitie. Niektoré algoritmy spracúvajú iba štatistiky založené na paketovom prenose (IP), kde iné zahŕňajú aj analógové merania ako je šum, hlasitosť, echo a skreslenie aby sa zvýšila presnosť a opakovateľnosť. Pretože ucho je analógové zariadenie a zvuk je analógový signál, je dôležité zahrnúť do výsledkov aj túto analýzu. MOS ohodnotené použitím IP aj analógových meraní omnoho presnejšie vystihuje subjektívny pocit z hovoru.

PESQ ITU T P.862

Vstupy tohto algoritmu sú analógové merania. PESQ algoritmus je založený na psycho-akustickom modeli navrhnutom na vyhodnotenie hlasovej kvality tak, ako ho vnímajú volajúci účastníci. Algoritmus porovnáva referenčný hlasový súbor s nahrávkou toho istého súboru po tom, čo bol prenesený sieťou alebo zariadením ktoré sa testuje, merajúc efekty jednosmerného prenosu (skreslenie, šum) na hlasovej kvalite. Pretože originál a skreslená vzorka sú časovo závislé a amplitúdovo normalizované pre porovnávanie, PESQ preto neberie do úvahy faktory ako sú oneskorenie, jitter, echo alebo utlmenie.



Obr. 2.5 – 1 Schéma fungovania PESQ algoritmu

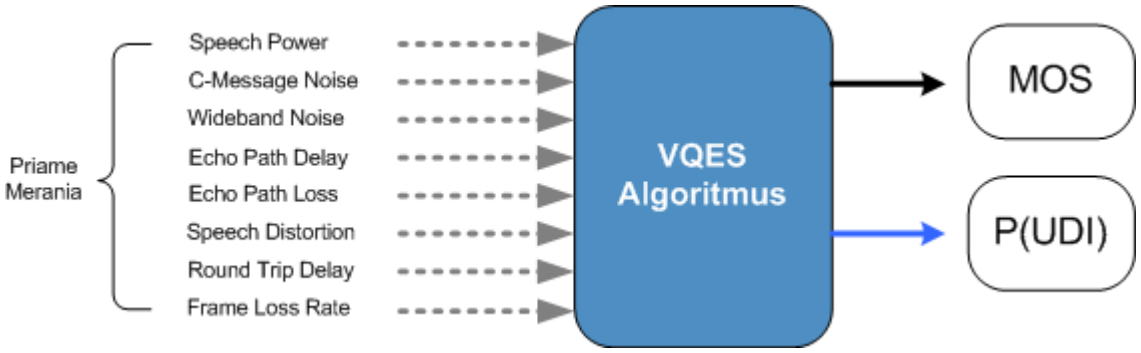
P.563 Listening MOS

P.563 je rozšírenie k PESQ P.562 algoritmu ktoré dovoľuje vypočítať MOS neintruzívne z aktuálnych hlasových vzoriek alebo nahrávok hovorov. P.563/PESQ kombinácia prepočítava MOS s použitím faktorov ako šum, echo, oneskorenie, potlačenie hlasu (clipping), utlmenie rámcov, strata paketov, kodek a typ siete.

Obr. 2.5 – 2 P.563 algoritmus

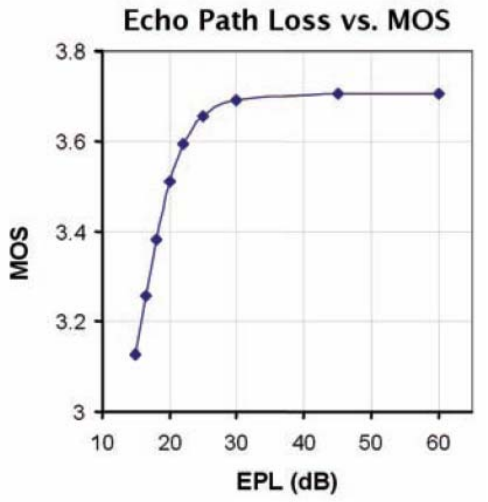
VQES Algoritmus

VQES algoritmus počíta s analógovými ako aj s paketovými faktormi hlasu. Je to sofistikovaný, na štatistikách založený algoritmus, ktorý vyhodnocuje spokojnosť koncového užívateľa a hlasovú kvalitu hovoru. VQES zahŕňa faktory ako je nízka hlasitosť (sila hovoru), šum, skreslenie, echo a oneskorenie. Tento algoritmus taktiež počíta tzv. P-UDI (Probability of Unusable, Difficult or Irritating calls (0-100%)), čo je metrika odzrkadľujúca úroveň frustrácie volajúceho s aktuálnou hlasovou kvalitou. VQES MOS je ovplyvnené echom, ale nie oneskorením, nakoľko P-UDI obidve tieto hodnoty zastrešuje.



Obr. 2.5 – 3 Schéma VQES algoritmu

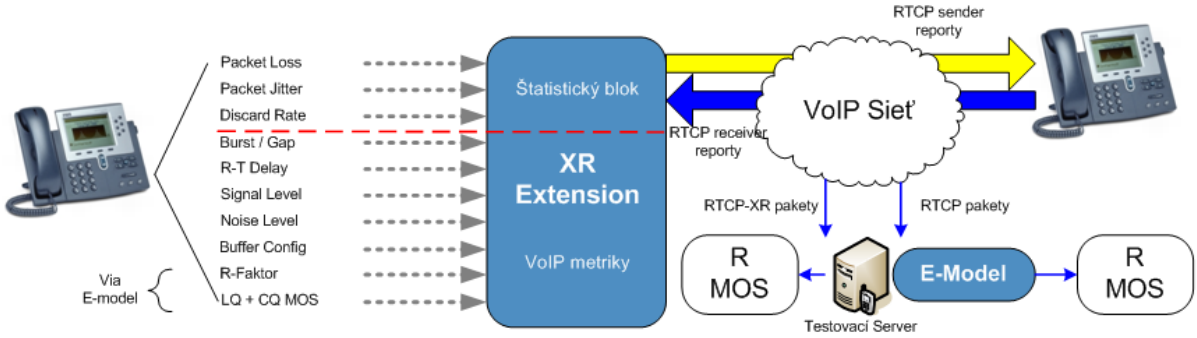
Nasledujúci graf ukazuje ako je MOS vypočítané pomocou VQES ovplyvnené echom. Echo je charakterizované tzv. Echo Path Loss (EPL) a Echo Path Delay (EPD), dvoma vzájomne prepojenými metrikami. Kombinácia týchto dvoch meraní ovplyvňuje celkové pôsobnie echa na vnem kvality prenášaného hlasu.



RTCP & RTCP-XR - IETF RFC-3611

Tento algoritmus počíta s analógovými ako aj s paketovými faktormi hlasu. RTCP ako som už spomínal je kontrolnou časťou špecifikácie RTP protokolu, ktorý sa používa na prenos hlasu a videa cez IP siete. Štandardné RTCP pakety obsahujú základné parametre kvality o prenose ako strata paketov, duplikácia, jitter, TTL paketu a obmedzenie počtu hopov. RTCP Extended Report (RTCP-XR) pakety prenášajú ešte dodatočné parametre kvality ako je round-trip-delay (RTD), sila signálu a šumu, R-faktor, MOS konverzácie a počuteľnosti a konfiguráciu jitter buffrov. RTCP avšak reprezentuje len prostriedok pre oznamovanie kvality prenosu.

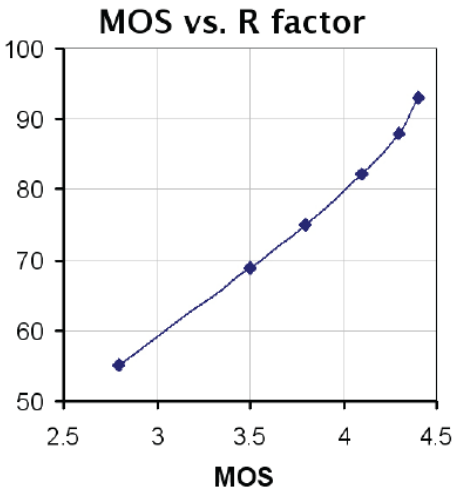
Koncové body relácie a sieťové prvky nahrávajú a prepočítavajú metriky prenášané v RTCP blokoch vysielača a prijímača (RTCP-SR/RR). Tieto informácie môžu byť potom použité kompatibilnými zariadeniami k dynamickému prispôsobovaniu konfigurácie aby sa tak optimalizovala kvalita prenosu. RTCP a RTCP-XR pakety sa taktiež používajú testovacími systémami na meranie a oznamovanie kvality služieb. Dodatočné parametre kvality obsiahnuté v XR paketoch sú vo všeobecnosti vypočítavané na koncových zariadeniach alebo testovacom vybavení použitím E model algoritmu, s použitím štandardných RTCP štatistík.



Obr. 2.5 – 4 Schéma vyhodnocovania MOS na základe RTCP-XR

E-Model G.107, 108, 109

E-model ohodnocuje konverzačnú kvalitu (CQ) a počuteľnú kvalitu (LQ) použitím R-faktoru, a môže byť prekonvertovaný na príslušnú hodnotu MOSCQ a MOSLQ ako aj na dve ďalšie agregované metriky – Good-or-Better (GoB) a Poor-or-Worse (PoW) ukazovatele. MOSCQ zahŕňa do výpočtov efekt oneskorenia a echa kým MOSLQ tieto ukazovatele nevyužíva. E-model bol pôvodne navrhnutý pre plánovanie sietí a testovanie kodekov – pre výpočty oneskorenia, echa a skreslenia ktoré vyplýva z kompresie a dekompresie hlasu pomocou kodekov. Rôzne firemné modifikácie zvýšili univerzálnosť a presnosť E-modelu tým, že sa do R-faktoru zahrnuli parametre špecifické pre VoIP ako IP a analógové štatistiky. E-model sa v praxi väčšinou používa iba s IP štatistikami a chýbajúce analógové štatistiky sú nahradené nominálnymi hodnotami.



V tejto práci sa odvolávam na MOS hodnoty ktoré sú vypočítané výhradne s použitím IP štatistík, avšak pre potreby mojich experimentov je táto presnosť dostačujúca. Konkrétny použitý algoritmus je potom stanovený konkrétnym meracím produktom.

3. Quality of Service (QoS)

3.1 Čo je QoS

Trendom dnešnej doby je pre sieťových dizajnérov postaviť tzv. multiservice sieť, schopnú prenášať všetky typy komunikácie - hlas, dáta a video pomocou paketovej architektúry. Požiadavka po stále väčšej šírke pásma je neutíchajúca a v poslednej dobe ešte zrýchľuje. Avšak zvýšená požiadavka po šírke pásma môže spôsobiť problémy s kvalitou, a to hlavne degradáciu časovo senzitívneho typu sieťovej prevádzky akou je hlas. Hlasové pakety nemajú také isté výhody ako dátové čo sa týka ich zahadzovania a následnej retransmisie.
QoS sa odkazuje na schopnosť siete poskytovať lepšie služby pre vybraný typ sieťovej prevádzky nad rozličnými podliehajúcimi prenosovými technológiami, zahrňujúc IP smerované siete, Frame Relay, ATM, Ethernet a 802.1 alebo Synchrónne optické siete (SONET/SDH). Vo všeobecnosti, QoS funkcie poskytujú lepšie a viac predvídateľné sieťové služby tak, že:

- Zlepšujú charakteristiky stratovosti
- Zabraňujú a menežujú sieťové zahltenie
- „Shapujú“ sieťovú prevádzku
- Nastavujú prioritu sieťovej prevádzky naprieč sieťou

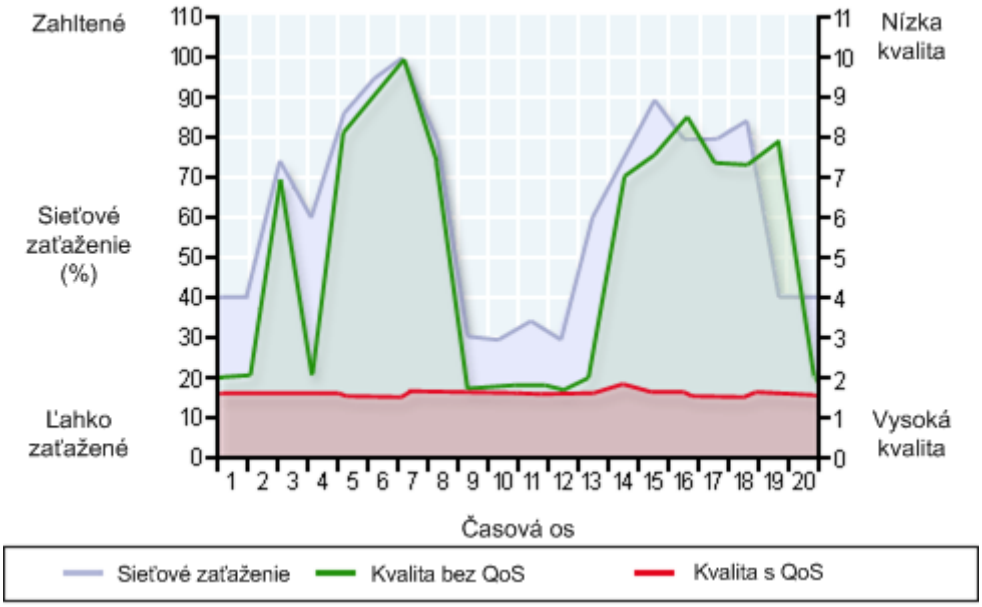
Konkrétne elementy QoS architektúry závisia od typu prenášaných informácií (hlas, dáta alebo video). Pre VoIP definuje QoS obmedzenia špecifické k prenosu hlasu ako sú delay (oneskorenie), delay variation alebo aj jitter (časové chvenie), packet loss (strata paketov) a tiež echo. V ďalšom texte sa budem kvôli jednoduchosti odvolávať na anglické preklady týchto parametrov.
Z nasadenia QoS do exitujúcich alebo nových sietí plynú isté výhody a to hlavne:

- Redukcia nákladov na prevádzku
- Vyššia výkonnosť siete
- Väčšia flexibilita, integrácia a kontrolovateľnosť
- Rýchlejšie nasadzovanie aplikácií a služieb

Keďže na internete ako aj v rôznych iných zdrojoch sa definície QoS líšia, vybral som jednu, ktorá sa mi zdala ako najvyhovujúcejšia.

Quality of service (QoS) sa odkazuje na výšku poskytovanej kvality služby danému užívateľovi. QoS je množina nastaviteľných nástrojov ktoré poskytujú selektívne služby pre sieťovú prevádzku a tým vytvárajú lepšie služby pre rôzne typy sieťovej prevádzky ako je napríklad aj hlas.

QoS môže zvýšiť šírku pásma pre časovo citlivé dáta a aplikácie, obmedziť šírku pásma pre nekritický sieťový prenos a poskytnúť konzistentnú sieťovú odozvu. Takýto scenár zvyšuje efektívnosť drahých sieťových pripojení. Bez týchto QoS mechanizmov by mohli nepodstatné aplikácie veľmi rýchlo vyčerpať sieťové zdroje na úkor dôležitejších alebo priam kritických aplikácií, čo by obmedzilo firemné/nefiremné procesy a tým aj produktivitu.
Na praktickú demonštráciu ako môžu QoS mechanizmy pomôcť pri prenose hlasu cez IP sieť uvádzam v prílohe 1 dva zvukové súbory – jedná sa o prenášaný telefonický hovor bez a s použitím QoS. Tu je možné počuť výraznú zmenu v kvalite hlasu. S QoS kontrolou použitou na VoIP dáta narástla dostupná šírka pásma a samotným paketom je daná prednosť pred súperiacimi paketmi iného typu. To viedlo k lepšej kvalite hlasového signálu.
Nasledujúci graf znázorňuje porovnanie hlasovej kvality bez QoS a s QoS v sieti.



Obr. 3.1 -1 Porovnanie kvality bez QoS a s QoS

V grafe je dátové zaťaženie linky zobrazené šedo-modrou čiarou. Hodnoty kvality s aj bez QoS sú vyznačené zelenou a červenou čiarou.

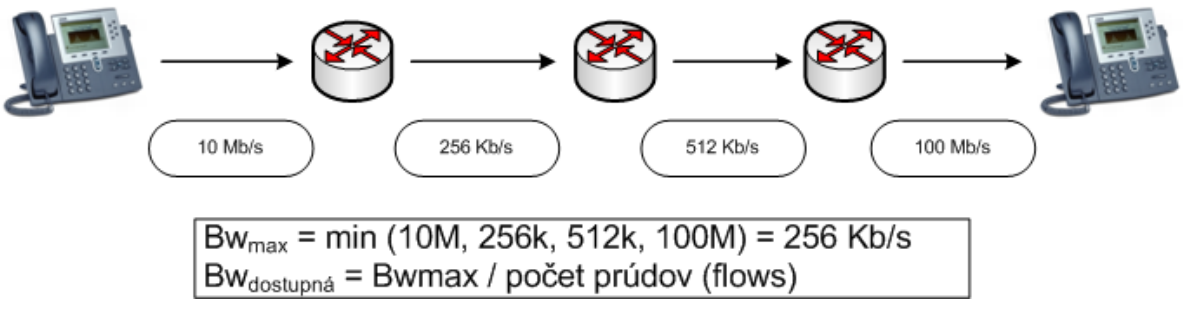
3.2 Faktory ovplyvňujúce VoIP QoS

Kvalitu hlasu vo VoIP priamo ovplyvňujú dva hlavné faktory:

- Dostupná šírka pásma
- Strata paketov - Loss
- Oneskorenie paketov – Delay
- Chvenie (Jitter)

Taktiež echo nepriamo ovplyvňuje kvalitu hlasu vo VoIP sieťach.

Dostupná šírka pásma - Maximálna dostupná šírka pásma je rovná šírke pásma najslabšej linky. Niekoľko prúdov dát súperí o tie isté sieťové zdroje a to má za následok, že daná konkrétna aplikácia má k dispozícii omnoho menej sieťových zdrojov. Schéma 3.2 - 1 ukazuje nezaťaženú sieť so štyrmi sieťovými uzlami medzi klientom a serverom. Každý uzol používa iný typ prenosového média a má inú šírku pásma. Maximálna dostupná šírka pásma na tejto trase sa teda rovná tej najslabšej linke. Výpočet dostupnej šírky pásma sa avšak komplikuje v prípade keď sieťou prenášame niekoľko prúdov dát. Kalkulácia zobrazená na obrázku je teda iba hrubým odhadom.



Obr. 3.2 – 1 Kalkulácia dostupnej šírky pásma

Strata paketov - Strata paketu nastáva keď je paket zahodený uzlom siete pretože ho nemohol preposlať na výstupné rozhranie. Existuje niekoľko dôvodov straty paketov:

- Zahltenie spôsobené frontami ktoré prečerpali svoj limit počtu paketov.
- Sieťovému uzlu došlo voľné miesto v buffri.
- Pamäťovým obmedzením sieťových uzlov.
- Tzv. Policing alebo riadenie, ktoré sleduje prúdy dát, zabezpečujúce aby sa vyhovelo istej hladine šírky pásma. Ak je šírka pásma prekročená, policing kontrola zahodí všetky pakety ktoré prekračujú tento limit.
- Porucha prenosovej linky

Strata paketov spôsobuje hlasové útržky a preskoky. Pretože v laboratóriu pracujem so zariadeniami spoločnosti Cisco, budem sa orientovať na riešenia, ktoré ponúka Cisco systems. Momentálne DSP procesory cisco (digital signal processor) a ich korekčné algoritmy môžu napraviť okolo 30 milisekúnd (msec) strateného hlasu. VoIP technológia Cisco zariadení používa 20 msec vzorky zvukového nákladu pre každý VoIP paket. Preto sa môže stratiť iba jeden paket počas nejakej doby aby bol algoritmus kodeku ešte efektívny.

Oneskorenie paketov - Oneskorenie paketov môže zapríčiniť buď degradáciu hlasovej kvality spôsobenou end-to-end latenciou alebo stratu paketov, ak je oneskorenie variabilné. Ak táto end-to-end latencia prekročí maximálnu hranicu (vo všeobecnosti sa za maximum považuje 250 msec), môže konverzácia nadobudnúť podobu komunikácie pomocou vysielačiek (jeden hovorí a na konci hlasovým povelom predá slovo druhej strane). Ak je oneskorenie paketov veľmi variabilné, nastáva tu riziko že na prijímacom konci pretečie tzv. jitter buffer. Eliminovanie výpadkov a oneskorení je ešte o to nevyhnutnejšie, ak sa rozhodneme cez IP prenášať aj fax a modemové dáta.

Ak sa nepodniknú kroky pre minimalizáciu straty a oneskorenia paketov, bude mať signál na vzdialenom konci siete veľmi úbohú kvalitu, vyplývajúc z veľkého celkového oneskorenia (veľké celkové oneskorenie je tiež nazývané „neprirodzená konverzácia“ alebo „satelitný efekt“ a znie podobne akoby bol človek zavretý v ozvenovej komore), alebo praská a strácajú sa slabiky kvôli strateným alebo oneskoreným paketom.

Aby sa zabezpečila vysoká kvalita hlasu v paketovo-orientovaných sieťach, musí sa aplikovať hneď niekoľko QoS techník. Výsledkom použitia týchto metód je minimalizácia straty, absolútneho oneskorenia ako aj variabilného oneskorenia (jitter) paketov ktoré prenášajú hlasové dáta cez sieť.

Ozvena (Echo) - Echo nie je problém špecifický pre prepínanie paketov. Taktiež ovplyvňuje kvalitu hlasu pri technológiách prepínaných okruhov. Sú dve príčiny vzniku echa :

- Sila vstupného signálu (gain - zisk) / sila výstupného signálu (attenuation - tlmenie)
- Nezodpovedajúca impedancia

Pre silu vstupu a výstupu, ak je sila vstupu hlasového portu na lokálnom smerovači príliš vysoká, druhá strana bude počuť hovorené útržky. Hlasový signál taktiež môže znieť rozostrene. Ak je sila vzdialeného výstupu príliš vysoká, môže volajúci počuť ozvenu od vzdialenej PBX (Private branch exchange).
Nezodpovedajúca impedancia zase spôsobí, že volajúci počuje svoju ozvenu zo vzdialenej strany. Pretože echo nie je špecifické pre VoIP, budem sa ďalej zameriavať len na faktory ako strata alebo oneskorenie paketov.

3.3 Strata a oneskorenie paketov

Preskúmaním následkov straty a oneskorenia paketov zistíme, prečo je použitie QoS potrebné vo všetkých sférach podnikových a iných sietí.

Kvalita siete - Hlasové pakety môžu byť zahodené ak:

- Kvalita siete nie je dostatočná (nesprávny návrh, nestabilné sieťové komponenty, zlé napájanie alebo nedostatočná šírka pásma)
- Sieť je zahltená (príliš veľa sieťovej prevádzky[hlas, dáta, video])
- V sieti je príliš veľa variabilného oneskorenia – jitter

Zlá kvalita siete môže viesť k často sa rušiacim reláciám kvôli strate fyzického alebo logického spojenia.

Zahltenie siete - Zahltenie siete môže viesť aj k zahadzovaniu paketov aj variabilnému oneskoreniu paketov. Zahadzovanie hlasových paketov v súvislosti so zahltením siete je zvyčajne zapríčinené plným prenosovým bufferom na výstupnom rozhraní. Ako sa vyťaženie siete približuje 100%, začínajú sa fronty obsluhujúce toto spojenie postupne plniť. Keď sa front naplní, pakety, ktoré sa pokúsia vstúpiť do plného frontu budú zahodené. Toto je celkom bežný jav a môže nastať všade – od internátnych prepínačov až po frame relay sieť poskytovateľa služieb.
Pretože zahltenie siete býva väčšinou sporadické a náhodné, taktiež oneskorenia spôsobené zahltením majú tendenciu sa meniť. Tieto variabilné oneskorenia sú spôsobené frontami na výstupných rozhraniach – časom ktorý paket strávi vo fronte, alebo v tzv. dejitter buffroch alebo je to spôsobené rozličnou veľkosťou paketov. Tieto scenáre oneskorenia sú rozobrané ďalej v texte.

Delay a Jitter - Oneskorenie je množstvo času, ktoré paket potrebuje na dosiahnutie svojho cieľa v sieti po tom čo bol vyslaný vysielačom. Tento časový údaj sa tiež nazýva „end-to-end“ oneskorenie a môže byť rozdelený na dve časti: fixné sieťové oneskorenie a variabilné sieťové oneskorenie. Jitter je v podstate delta, alebo rozdiel v end-to-end hodnotách dvoch hlasových peketov hlasového streamu.

3.3.1 Typy sieťového oneskorenia

Následky straty paketov a oneskorenia môžeme klasifikovať na dva typy:

- Fixné sieťové oneskorenie
- Variabilné sieťové oneskorenie

Fixné sieťové oneskorenie **-** Fixné sieťové oneskorenie pozostáva z troch komponentov:

- Oneskorenie vplyvom šírenia – Propagation delay
- Serializačné oneskorenie – Serialization delay
- Oneskorenie vplyvom spracovania – Processing delay

Propagation Delay - Oneskorenie vplyvom šírenia, je oneskorenie signálu medzi vysielajúcim a prijímajúcim koncovým bodom a je založené na celkovej vzdialenosti medzi zdrojom a cieľom. Všeobecná hodnota oneskorenia vplyvom šírenia je 6 mikrosekúnd (µsec) na kilometer (km). Reálne výpočty sa pohybujú okolo hodnoty 6.3 µsec/km.

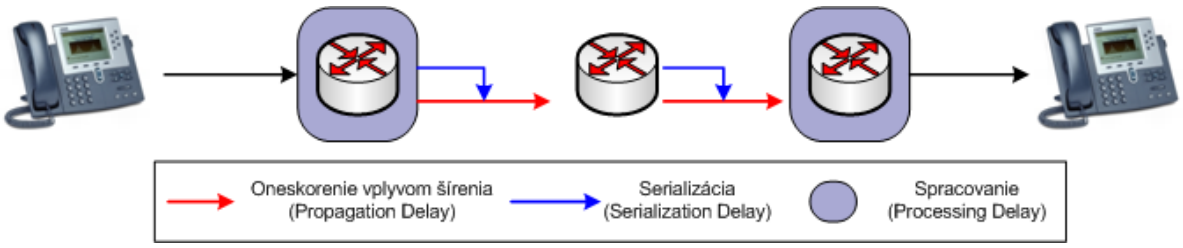
Serialization delay - Oneskorenie vplyvom serializácie je výsledkom umiestňovania bitov na sieťový spoj. Čím vyššia rýchlosť linky, tým menej času je potrebné aby sa na ňu umiestnili jednotlivé bity. Teda čím vyššia rýchlosť, tým menšie serializačné oneskorenie. Napríklad umiestnenie jedného bajtu na 64 Kb/s linku trvá 125 µsec. Pre porovnanie, umiestnenie toho istého bajtu na OC-3 linku (optika – 155 Mb/s) trvá asi 0.5 µsec.

Serializačné oneskorenie je konštantná funkcia pozostávajúca z rýchlosti linky a veľkosti paketu. Čím väčšia veľkosť paketu a nižší takt linky, tým väčšie serializačné oneskorenie. Toto oneskorenie nie je variabilné (alebo nepredvídateľné) v porovnaní napríklad s oneskorením vo frontoch, kde toto oneskorenie môže byť aj nulové aj nenulové. Môžeme ho vždy presne určiť – pre 64 Kb/s linku a 80 bajtový fragment bude oneskorenie presne 10ms.
Serializačné oneskorenie je faktor, ktorý nás zaujíma hlavne na pomalých linkách, čiže s rýchlosťou menšou ako 1 Mb/s.

Processing delay - Oneskorenie vplyvom spracovania môže byť rozdelené na:

- Oneskorenie vplyvom kódovania, kompresie, dekompresie a dekódovania, ktoré závisí od použitého algoritmu. Tieto funkcie môžu byť vykonávané buď hardvérovo alebo softvérovo. Použitím špecializovaných zariadení ako je DSP sa dramaticky zlepšuje kvalita a redukuje oneskorenie spojené s rôznymi hlasovými kompresnými schémami.
- Tzv. Paketizačné oneskorenie (Packetization delay) je proces zadržiavania digitálnych hlasových vzoriek po určitú dobu kým sa ich nenazbiera toľko aby sa dali umiestniť do nákladu paketu alebo bunky. Aby sa zredukovalo nadmerné oneskorenie v niektorých kompresných schémach, je možné posielať čiastočné pakety.

Nižšie je obrázok zobrazujúci tieto tri komponenty fixného sieťového oneskorenia a kde nastáva vo vzťahu s end-to-end prenosom.



Obr. 3.3.1 – 1 Komponenty a miesta fixného sieťového oneskorenia

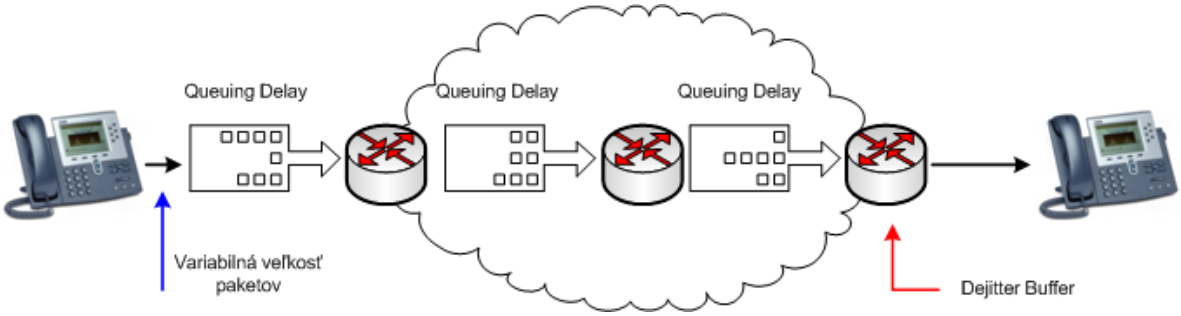
Poznámka: Fixné sieťové oneskorenie by malo byť preskúmané ešte v úvodnej fáze návrhu VoIP siete. ITU štandard G.114 určuje, že maximálne 150 msec jednosmerné oneskorenie je ešte postačujúce pre vysokú kvalitu hovoru.

Pre lepšiu demonštráciu si vezmem G.729A hlasový kodek. Ten má oneskorenie vplyvom šírenia už spomínaných 6.3 µsec/kilometer, 25 msec potrebných na zakódovanie (dva 10 msec framy a 5 msec look-ahead) a ďalších 20 msec paketizačného oneskorenia. Potom môžeme povedať, že kodek G.729A má fixné oneskorenie 45 msec (plus oneskorenie vplyvom šírenia čo je 6.3 µsec/kilometer).

Variabilné sieťové oneskorenie - Nasledujúce tri faktory ovplyvňujú variabilné sieťové oneskorenie:

- Oneskorenie následkom čakania vo fronte – Queuing delay
- Dejitter buffre – Jitter
- Variabilná veľkosť paketov

Na obrázku nižšie sú zobrazené tieto typy oneskorenia, všetky sú variabilné a sú viac kontrolovateľné.



Obr. 3.3.1 – 2 Variabilné sieťové oneskorenie

Queuing Delay - Zahltené výstupné fronty na sieťových rozhraniach sú najväčším zdrojom variabilného oneskorenia. Queuing oneskorenie nastáva, keď musí paket čakať kým sa na linku umiestnia pakety pred ním vo fronte. Tento čakací čas je štatisticky založený na príchode dát, preto čím viacej zdrojov dát, tým ľahšie môže nastať na kanáli zahltenie.

Dejitter Buffers - Pretože sieťové zahltenie môže nastať v ktorýkoľvek časový okamih, buffre sa môžu naplniť takmer okamžite. Toto okamžité naplnenie môže viesť k rozdielu v časoch oneskorenia medzi paketmi toho istého hlasového prúdu. Tento rozdiel sa nazýva Jitter a je to rozdiel medzi časom v ktorom sa očakáva, že paket dorazí a skutočným časom dorazenia paketu. Aby sa kompenzovala tieto delty oneskorenia v hlasových paketoch v danej relácii, používajú VoIP koncové zariadenia tzv. dejitter buffre (alebo jitter buffre) ktorých úlohou je premeniť tieto delty oneskorenia na konštantnú hodnotu, tak aby sa koncový zvuk mohol plynule prehrať.
Jitter buffer je používaný aby dočasne zdržal pakety aby sa vyrovnali hodnoty oneskorenia medzi jednotlivými paketmi. Nastavenie príliš malých buffrov spôsobí časté pretečenie a stratu dát. Naopak nastavenie príliš veľkých buffrov spôsobí nadmerné oneskorenie. Úlohou dejitter buffrov je teda premena variabilného oneskorenia na fixné oneskorenie.

Koncové VoIP zariadenia Cisco používajú DSP algoritmy ktoré majú adaptívny jitter buffer v rozsahu od 20 do 50 milisekúnd. Presná veľkosť sa potom nastavuje priebežne podľa očakávaného oneskorenia hlasových paketov. Tieto algoritmy skúmajú časové pečiatky v hlavičke RTP protokolu v hlasových paketoch, vypočítajú očakávané oneskorenie a prispôsobia podľa toho jitter buffer.

Variabilná veľkosť paketov - Variabilné oneskorenie je tiež závislé na veľkosti práve spracovávaného paketu: väčším paketom trvá viac než sa prenesú ako malým paketom. Front kombinujúci malé a veľké pakety zaznamená rôzne časy oneskorenia.

3.4 Metódy implementácie QoS

V zásade rozlišujeme tri metódy implementácie Quality of Service v paketových sieťach. A to :

- Best-Effort – Internet v jeho počiatku, bol navrhnutý pre Best-Effort systém doručovania. Čiže bez garancie doručenia paketov. Aj v dnešnej dobe Internetu je tento systém doručovania najrozšírenejší.
- Model Integrated Services - Uvedený ako náhrada pre Best-Effort doručovanie. Funguje tak, že rezervuje nejakú časť sieťových zdrojov pre aplikácie, ktoré potrebujú garanciu šírky pásma a oneskorenia. Model Intserv (Integrated Services) očakáva, že aplikácie budú svoje požiadavky sieti signalizovať. Na to sa využíva protokol Resource Reservation Protocol (RSVP), ktorý signalizuje QoS požiadavky sieti.
- Model Differentiated Services – Bol vytvorený pre poskytnutie väčšej škálovateľnosti pri poskytovaní QoS IP paketom. Hlavným rozdielom je, že sieť rozozná paket (nie je potreba signalizácie) a poskytne mu zodpovedajúce služby alebo starostlivosť. Dnešné IP siete môžu použiť všetky tri modely súčasne.

3.4.1 IntServ Model

Internet Engineering Task Force (IETF) je organizácia zodpovedná za štandardizáciu protokolov a sieťových architektúr ako ich poznáme z dnešného internetu. IETF je tu preto aby vytvárala štandardy, ktoré by dovoľovali interoperabilitu výrobkov rôznych výrobcov. Podobným spôsobom vznikol aj model Ingerated Services (IntServ, RFC1663), ktorý bol navrhnutý ako štandard spolu s Resource Reservation Protokolom (RSVP) ako prostriedkom pre signalizáciu QoS požiadaviek sieti.

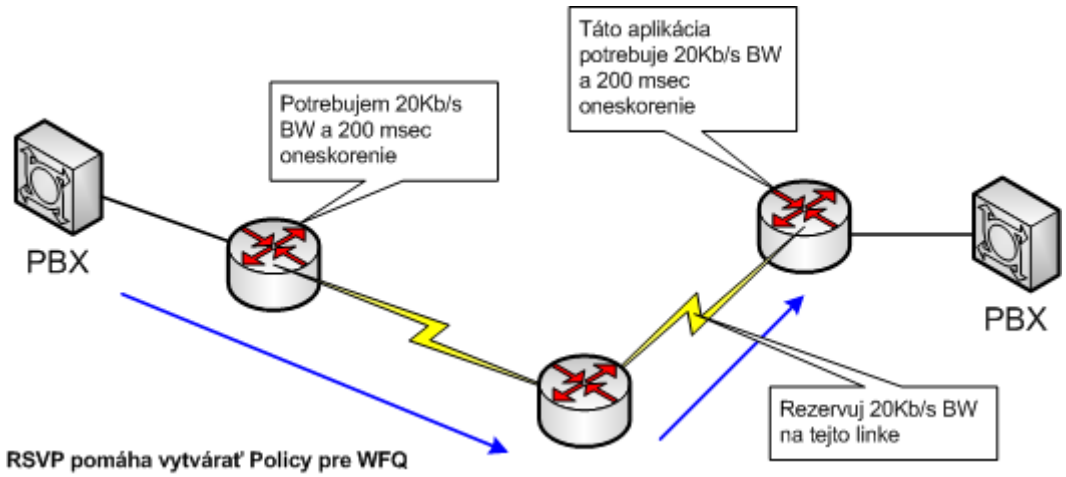
IntServ model stavia na predpoklade, že sieťové zdroje ktoré sú nám k dispozícii majú byť explicitne rezervovateľné, čím sa môže zabezpečiť dostatočná kvalita tej ktorej služby. Takto každé jednotlivé spojenie alebo prúd dát dostane striktne pridelené, ktoré z dostupných zdrojov môže použiť. Preto musí každý smerovač na ceste paketu uchovávať stavové informácie daných tokov a taktiež rozhodovať ktoré dátové toky môžu využiť ktoré QoS sieťové zdroje. Toto sa deje práve s použitím protokolu RSVP.

Stavebné bloky IntServ modelu sú:

- Rezervácia zdrojov (Resource Reservation) – používa sa na identifikáciu aplikácie (prúdu dát) a signalizuje do siete požiadavky, či je dosť pre ňu dostatok sieťových zdrojov. Rezervácia je implementovaná použitím RSVP.
- Call Admission Control (CAC) – používa sa na určenie, či daná aplikácia (prúd dát) môže dostať pridelené žiadané zdroje. Implementácia je buď lokálna na smerovačoch alebo je centralizovaná na centrálnom serveri. Protokol používaný pre centrálnu správu je tzv. COPS (Common Open Policy Service) štandardizovaný IETF v RFC 2748 a jeho použitie s RSVP je definované s RFC 2749.

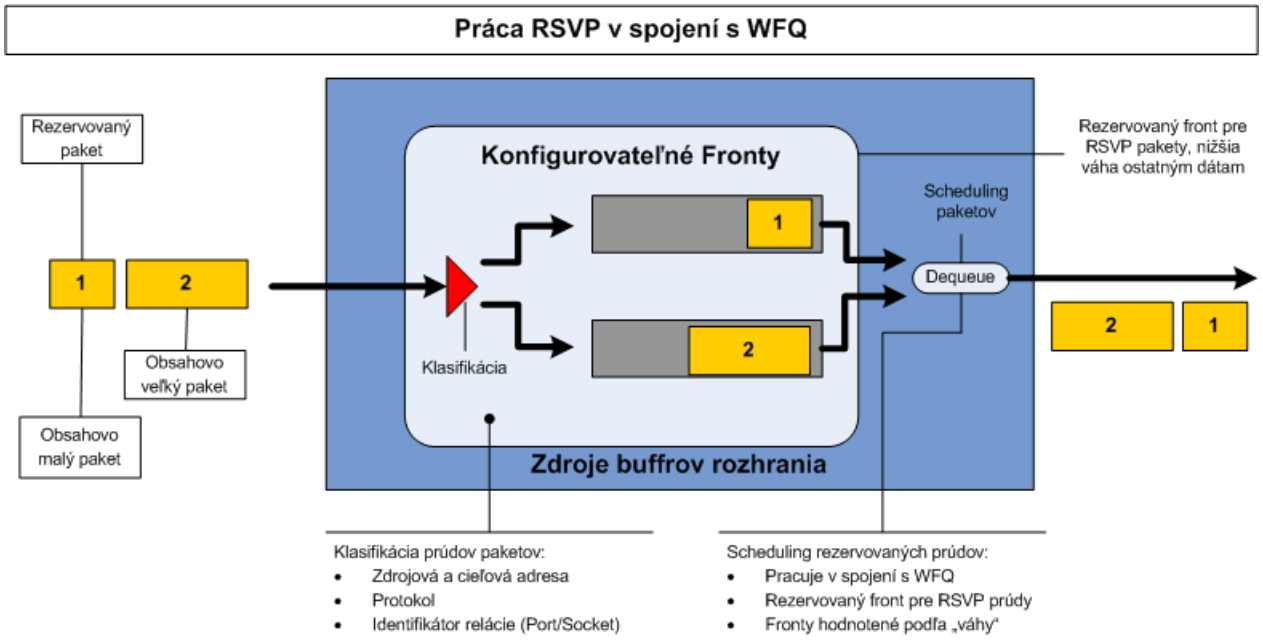
Resource Reservation Protocol (RSVP) - RSVP je jediná forma end-to-end QoS (CAC) momentálne k dispozícii pre VoIP. Keď sa nadväzuje hovor sú posielané správy cez sieť a zase späť k pôvodcovi, vyžadujúce istú šírku pásma (okrem iných parametrov). Ak je táto šírka pásma dostupná po celej trase, hovor môže pokračovať a fronty po celej dĺžke trasy sa modifikujú tak, aby zabezpečili rezervovanie daných zdrojov. Ak ale daná šírka pásma nie je v sieti dostupná, mal by sa hovor ukončiť (realita je však taká, že

v súčasných softvérových implementáciách by hovor aj napriek tomu pokračoval, preto ja vidím RSVP ako spôsob prispôsobovania frontov ale nie naozajstným dosiahnutím CAC). Nasledujúca schéma popisuje činnosť RSVP.



Obr. 3.4.1 – 1 Schéma činnosti RSVP protokolu

RSVP pracuje v spojení s WFQ (Weighted Fair Queuing) takýmto spôsobom: keď nejaký smerovač v sieti súhlasí s RSVP požiadavkou, vytvorí nový front pre dáta s požadovateľovou zdrojovou adresou a alokuje mu dostatok zdrojov a dá absolútnu prioritu nad rozhraním. Takým istým spôsobom ako rezervácia RTP, ak rezervovaná šírka pásma nie je úplne vyčerpaná, môže sa použiť pre iný typ sieťovej prevádzky.



Obr. 3.4.1 – 2 RSVP v spojení s WFQ radením do frontov

Pre potreby VoIP sietí môže byť RSVP veľmi užitočnou metódou. Zabezpečuje, že sa frontom priradí adekvátna priorita tak, aby dáta v ňom neboli zahodené a dáta v ňom utrpeli minimálne oneskorenie a jitter. Problémy a nedostatky RSVP súvisia s jeho správou: RSVP musí byť nakonfigurované na každej smerovači a rozhraní v sieti s údajom koľko percent šírky pásma môže byť rezervovateľné. Navyše, veľkosť

rezervovanej šírky pásma je stanovené natvrdo na 24 K pri použití G.729 kompresie a 84K pre G.711. Nepočíta teda ani s kompresiou RTP hlavičky ani s VAD. Taktiež v momentálnych cisco implementáciách by hovor bol nadviazaný aj v prípade nedostatku zdrojov (RSVP správy sa pošlú až po počiatočnej signalizácii hovoru). Spolu tieto nevýhody robia RSVP v dnešných VoIP sieťach veľmi neefektívnym.

3.4.2 DiffServ Model

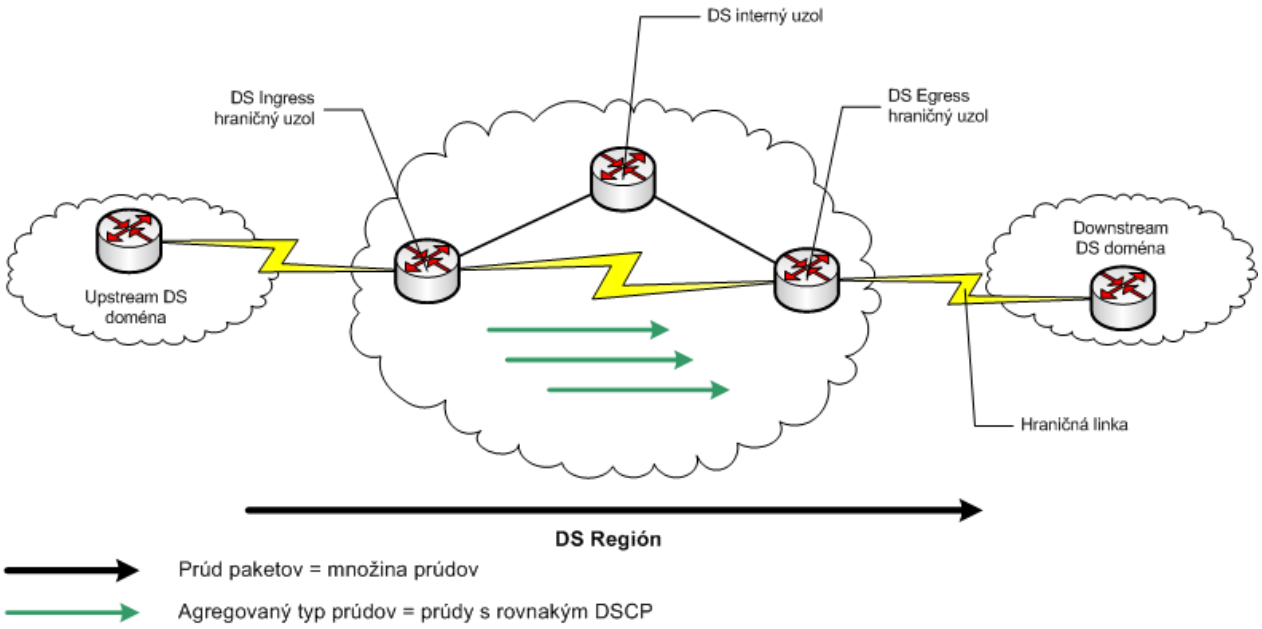
QoS model podľa DiffServ bol vytvorený IETF a je špecifikovaný v RFC 2474. V základe pracuje presne opačne ako v prípade modelu IntServ. Rozdiel je v tom, že sa vopred nerezervujú žiadne sieťové zdroje ako je to v prípade IntServ, ale pakety sú počas svojho prenosu priraďované do prenosových tried. Prenosové triedy sú identifikované hodnotou tzv. DiffServ Code Point – DSCP (DSCP nahrádza IP Precedenciu v ToS poli IP hlavičky). Hlavným cieľom DiffServ modelu je poskytnúť škálovatelnosť a podobnú úroveň QoS v porovnaní s IntServ modelom, bez toho aby sa to muselo robiť na základe jednotlivých prúdov dát. Sieť jednoducho rozozná triedu (nie aplikáciu) a aplikuje vhodný PHB (per hop behavior) QoS mechanizmus.

PHP sú metódy diferencovania IP paketov a tieto radia IP pakety do ich zodpovedajúcich prenosových tried (CoS – Class of Service). Z PHB hodnôt sa teda určia triedy podľa ktorých sa dané pakety potom následne spracujú a to na základe označenia v ToS poli (DSCP značka) IP hlavičky. Smerovač získa hodnotu CoS cez tabuľky v ktorých sa každej DSCP hodnote priraďuje PHB hodnota.

Pretože každej DSCP hodnote zodpovedá jedna PHB hodnota, sú tieto priradenia jednoznačné a určujú správanie sa smerovačov pri forwardovaní jednotlivých IP paketov sieťou.

DiffServ model popisuje služby a dovoľuje v takejto sieti použitie viacej užívateľom definovaných služieb. Tieto služby sú radené do tried. Trieda môže byť identifikovaná ako jedna aplikácia alebo, a to vo väčšine prípadov, môže byť identifikované zdrojovou alebo cieľovou IP adresou. Účelom je aby sieť rozoznala triedy bez toho aby dostávala nejaké požiadavky od aplikácií. Toto umožňuje aby boli QoS mechanizmy aplikovateľné aj na tie aplikácie, ktoré nemajú RSVP funkcionalitu, čo je prípad 99% aplikácií ktoré používajú IP.

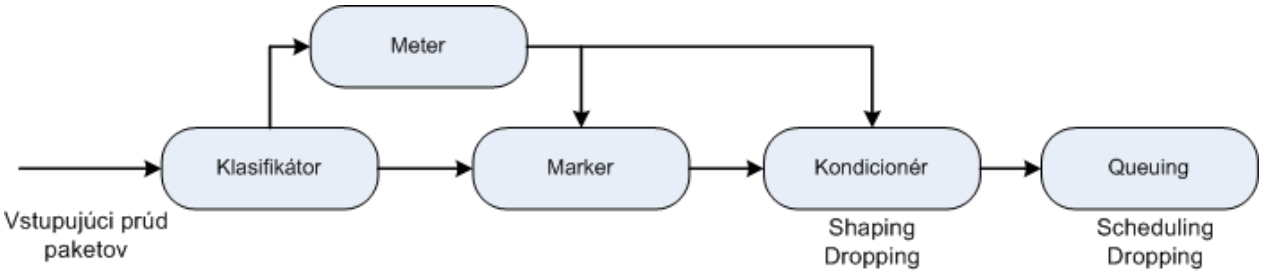
K tomuto účelu sa musia hlasové pakety v sieťových uzloch IP siete označiť (DSCP / IP Precedence marking). Toto sa udeje na smerovačoch tzv. Diffserv domény, ktorá predstavuje do seba uzavretý administratívny celok. Viacero DiffServ domén potom tvorí DiffServ región. Nasledujúca schéma znázorňuje topológiu DiffServ modelu.



Obr. 3.4.2 – 1 Topológia DiffServ modelu

DS (DiffServ) doména pozostáva z DS hraničných uzlov a DS interných alebo vnútorných uzlov. Hraničné DS uzly prepájajú DS doménu s inými DS alebo aj nie DS doménami, kým DS vnútorné uzly sa iba pripájajú na iné DS vnútorné alebo hraničné uzly v rámci jednej domény. Tak hraničné ako aj vnútorné uzly musia byť schopné aplikovať vhodné PHB paketom založeným na DS code point, v inom prípade by mohlo nastať nepredvídateľné chovanie siete.
DS hraničné uzly sa pre dáta v sieti chovajú aj ako DS ingress aj ako DS egress uzol. Sieťové pakety vstupujú do DS domény v uzle DS ingress a opúšťajú túto doménu v uzle DS egress. DS ingress uzol je zodpovedný za zabezpečenie že všetky dáta vstupujúce do DS domény budú spracované podľa tzv. Traffic Conditionig Agreement (TCA) medzi ním a druhou doménou ku ktorej je ingress uzol pripojený. Aby sa povolili služby ktoré sa tiahnu cez niekoľko DS domén v nejakom DS regióne musí hraničná DS doména vystaviť tzv. Service Level Agreement (SLA) ktorý definuje (buď explicitne alebo implicitne) TCA. TCA potom špecifikuje ako sa prenášané dáta budú spravovať na hraničných uzloch pri prechode z jednej domény do druhej.

Základná funkcia smerovačov vezme paket prijatý na vstupnom rozhraní, spraví nejaký typ rozhodnutia čo s paketom spraví a potom ho posunie ďalej (forwarding) cez výstupné rozhranie. Avšak dnešné schopnosti smerovačov sú o čosi sofistikovanejšie.
IP QoS mechanizmy Cisco IOS môžu vykonávať rôzne typy akcií Diffserv pracuje podľa nasledovnej schémy a používa pritom nižšie uvedené QoS metódy:



Obr. 3.4.2 – 2 Algoritmus práce DiffServ modelu

DiffServ QoS Metódy:

Klasifikácia – väčšina QoS mechanizmov podporuje viacero tried. Existuje niekoľko klasifikačných nástrojov pre rôzne QoS mechanizmy (napríklad Access listy, smerovacie mapy alebo mapy tried). Každý triedovo-orientovaný QoS mechanizmus musí podporovať niektorý typ klasifikácie.

Metering – niektoré mechanizmy merajú množstvo sieťovej prevádzky na sieti a podľa týchto informácií sa potom zvolí adekvátna akcia (napríklad obmedzenie priepustnosti, shaping alebo sheduling).

Dropping – niektoré mechanizmy sa používajú na zahadzovanie paketov. Vyberie sa nejaká schéma pre zahadzovanie paketov rozdielna oproti bežnému tail-dropu pri preplnení fronty. Jedným z takýchto mechanizmov je WRED (Random early detection).

Policing – niektoré mechanizmy sú používané na limitovanie sieťovej prevádzky na základe údajov z meteringu tak, že pakety ktoré sú nad rámec stanoveného limitu sa zahodia. (napríklad CAR – Commited Access Rate)

Shaping - niektoré mechanizmy sú používané na limitovanie sieťovej prevádzky na základe údajov z meteringu tak, že pakety ktoré sú nad rámec stanoveného limitu sa oneskoria. (napríklad GTS)

Marking – niektoré mechanizmy majú schopnosť značkovať pakety na základe klasifikácie alebo meteringu. (napríklad CAR alebo class-based marking, pomocou IP precedencie alebo DSCP).

Queuing – niektoré mechanizmy sú používané pre radenie do frontov na výstupných rozhraniach. (napríklad CQ, PQ, WFQ, CBWFQ alebo IP RTP Priority)

Forwarding – existuje niekoľko podporovaných forwarding mechanizmov. (Process switching, fast switching alebo Cisco express forwarding)

3.4 QoS metódy špecifické pre VoIP

Kvalita hlasu je iba taká dobrá ako kvalita najslabšej linky v sieti. Ako som už spomenul, strata paketov, oneskorenie a jitter – všetko toto prispieva k degradácii kvality prenášaného hlasu. A pretože zahltenie siete – alebo presnejšie, trvalé zaplnenie buffrov - môže nastať kedykoľvek, v ktorejkoľvek časti siete, je sieťová kvalita vecou end-to-end dizajnu siete. Nástroje QoS, ktoré využívam k zabezpečeniu dostatočnej kvality služby VoIP, sú sadou mechanizmov ktoré zvyšujú kvalitu hlasu prenášaného cez dátové siete tým, že znižujú počet zahodených paketov počas sieťového zahltenia a minimalizujú tak fixné ako aj variabilné oneskorenie.

VoIP QoS nástroje môžeme rozdeliť do troch kategórií:

- Klasifikácia – Classification
- Radenie do Front – Queuing
- Dimenzovanie linky – Network provisioning

Klasifikácia - Klasifikácia je označovanie paketov alebo dátových tokov určitou prioritou. Toto označovanie vytvára záväzok a ten sa musí dodržiavať. Klasifikácia by sa mala odohrávať na kraji siete, najlepšie v rámci koncových zariadení ak je to možné.

Dôležitosť paketov môžeme označiť použitím Layer 2 class-of-service (CoS) nastavenia v User Priority bitoch 802.1p časti 802.1Q hlavičky, alebo tiež pomocou IP Precedence / differentiated services code point (DSCP) bitov v type-of-service (ToS) bajte IPv4 hlavičky.
RTP pakety všetkých IP telefónov by mali byť označené s hodnotou CoS = 5 pre layer 2 802.1p nastavenie a IP Precedence = 5 pre layer 3 nastavenie. Dodatočne by mali byť všetky kontrolné pakety tiež označené a to s Layer 2 CoS hodnotou rovnou 3 a Layer 3 ToS hodnotou rovnou tiež 3.
Takéto použitie IP Precedence na značenie sieťovej prevádzky je iba prechodným krokom pokým všetky IP zariadenia nebudú podporovať DSCP.

Nasledujúca tabuľka zobrazuje vzťahy medzi hodnotami pre CoS, IP Precedence a DSCP.

Layer 2 Class of Service	IP Precedence	DSCP
CoS 0	Routine (IP Precedence 0)	0–7
CoS 1	Priority (IP Precedence 1)	8–15
CoS 2	Immediate (IP Precedence 2)	16–23
CoS 3	Flash (IP Precedence 3)	24–31
CoS 4	Flash-override (IP Precedence 4)	32–39
CoS 5	Critical (IP Precedence 5)	40–47
CoS 6	Internet (IP Precedence 6)	48–55
CoS 7	Network (IP Precedence 7)	56–63

Tab. 3.4 – 1 Vzťah medzi CoS, IP Precedence a DSCP

Radenie do front (Queuing) - Radenie do front – Queuing, priradí paket alebo tok dát k jednému alebo viacerým frontom (v závislosti na klasifikácii) pre náležité zaobchádzanie s tým ktorým typom dát. Akonáhle sa dáta, hlas a video začnú umiestňovať do tej istej fronty, je vysoká pravdepodobnosť že nastane vysoká strata paketov a variabilné oneskorenie. Použitím viacerých frontov na výstupnom rozhraní (namiesto použitia iba jedného pre všetok typ dát) a oddelenie dát od hlasových paketov robí chovanie siete oveľa predvídateľnejším.

Network provisioning - Dimenzovanie siete pozostáva z presného výpočtu požadovanej šírky pásma pre použitie voice over wide-area-network (WAN), všetok dátový tok, akékoľvek video aplikácie a nutnú správu linky – overhead, ako sú napr. smerovacie protokoly.
Je dôležité si uvedomiť, že všetka aplikačná sieťová prevádzka (hlas, dáta, video) by sa v súčte mala rovnať maximálne 75% navrhnutej celkovej šírky pásma. Zvyšných 25% by malo byť použitých na sieťovú réžiu ako sú smerovacie protokoly.

3.5 Komponenty QoS pre VoIP

Vo všeobecnosti je návrh akejkoľvek siete zložitým procesom, kde sa musí brať do úvahy jej topológia, šírka pásma liniek a typ prevádzky ktorý budeme prenášať a z týchto údajov potom analyticky odvodiť ktoré QoS funkcie manažmentu sieťovej prevádzky budú potrebné a ktoré metódy z každej funkcie použiť.

Ďalej sa budem venovať niektorým QoS metódam použiteľných pre VoIP a ich kombináciám, a to sú:

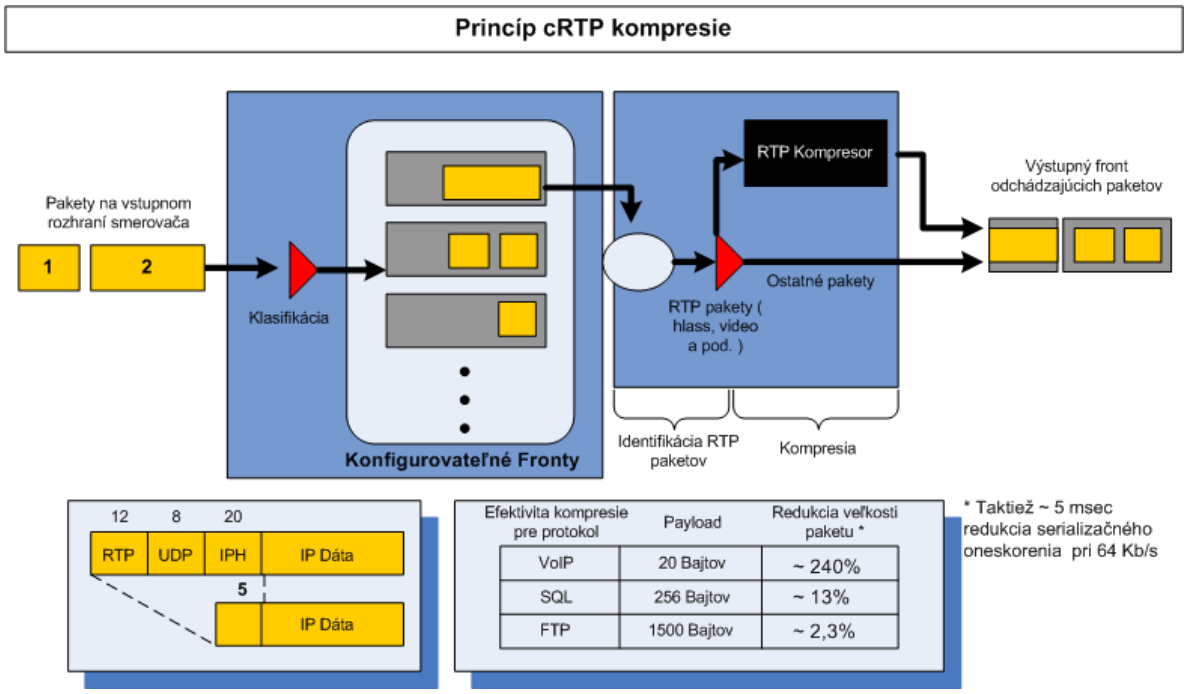
- Kompresia
- Marking
- Radenie do front – Queuing
- Traffic shaping a Policing
- Fragmentácia

3.5.1 Kompresia

Ako som už spomenul v úvode v rámci kodekov, nekomprimovaný hlas (G.711 PCM) vyžaduje na prenos 64 Kb/s pred pridaním dodatočného sieťového overheadu. Vo VoIP sieti vyžaduje takýto hovor až 84 Kb/s sieťových zdrojov. Preto VoIP brány alebo koncové zariadenia komprimujú daný hlasový signál pomocou kodekov ako je napr. G.729 na hodnotu okolo 8 Kb/s.
Kompresiou redukujeme potrebnú šírku pásma pre každý hovor čo v konečnom dôsledku zvyšuje počet možných hovorov na danej linke.

Dodatočné QoS funkcie v spojení s kompresiou - Okrem výberu vhodného kodeku sú k dispozícii aj iné QoS funkcie v rámci kompresie.

- **Voice activity detection (VAD**) – detekcia či dotyčný práve hovorí alebo nie. Ak je povolené VAD na hlasovom porte, tak ticho nie je sieťou prenášané, iba hovorené slovo. Kvalita zvuku sa tým síce trochu degraduje ale zato spojenie spotrebuje omnoho menej sieťových zdrojov. VAD môže zredukovať prenášané množstvo signálu o faktor závislý na obsahu ticha v pôvodnom signále, pre normálnu konverzáciu to znamená približne 35 % úspora.
- **Veľkosť nákladu paketu** – ďalším dôležitým faktorom pri určovaní šírky pásma na hlasový kanál je veľkosť hlasového nákladu (voice payload) v každom pakete. Táto možnosť pribudla v Cisco IOS® Release12.0(5)T kde sa počet bytov v náklade mohol špecifikovať ako časť výberu kodeku a môže mať dramatický dopad na vyťaženie linky, osobitne ak nie je použité CRTP. Vo všeobecnosti je veľkosť nákladu konfigurovateľná s krokom po 10 bajtoch až k maximu 240 bajtov, ale táto hodnota závisí od použitého kodeku.
- **Compressed RTP (CRTP)** – nutné konfigurovať na každej linke osobitne. Funguje tak, že sa vezme Real Time Transport Protocol/User Datagram Protocol (RTP/UDP) / IP hlavička hlasového paketu a skomprimuje sa z pôvodných 20 bajtov na približne 2 bajty. Pri počte prenesených paketov v rámci hovoru je úspora jasne viditeľná.



Obr. 3.5.1 – 1 Princíp fungovania cRTP

Nasledovná tabuľka ukazuje celkové nároky na šírku pásma pre dva konkrétne kodeky a čo je viac zaujímavé, taktiež overhead spojený s kodekmi a tiež Layer 2 overhead. Je tu znázornená aj redukcia veľkosti hlavičky pri použití CRTP.

Kodek	Hlasová šírka pásma (kb/s)	Veľkosť zakódovaného rámca (B)	Rámcov vo VoIP pakete	Veľkosť IP hlavičky (B)	Layer 2 technológia	Layer 2 veľkosť hlavičky (B)	Celková šírka pásma
G.711	64	80	2	40	Ethernet	14	85.6
G.711	64	80	2	40	MLP	6	82.4
G.711	64	80	2	2(CRTP)	MLP	6	67.2
G.729	8	10	2	40	Ethernet	14	29.VI
G.729	8	10	2	40	MLP	6	26.IV
G.729	8	10	2	2(CRTP)	MLP	6	11.II

Tab. 3.5.1 – 1 Porovnanie jednotlivých kodekov a úspora pri cRTP

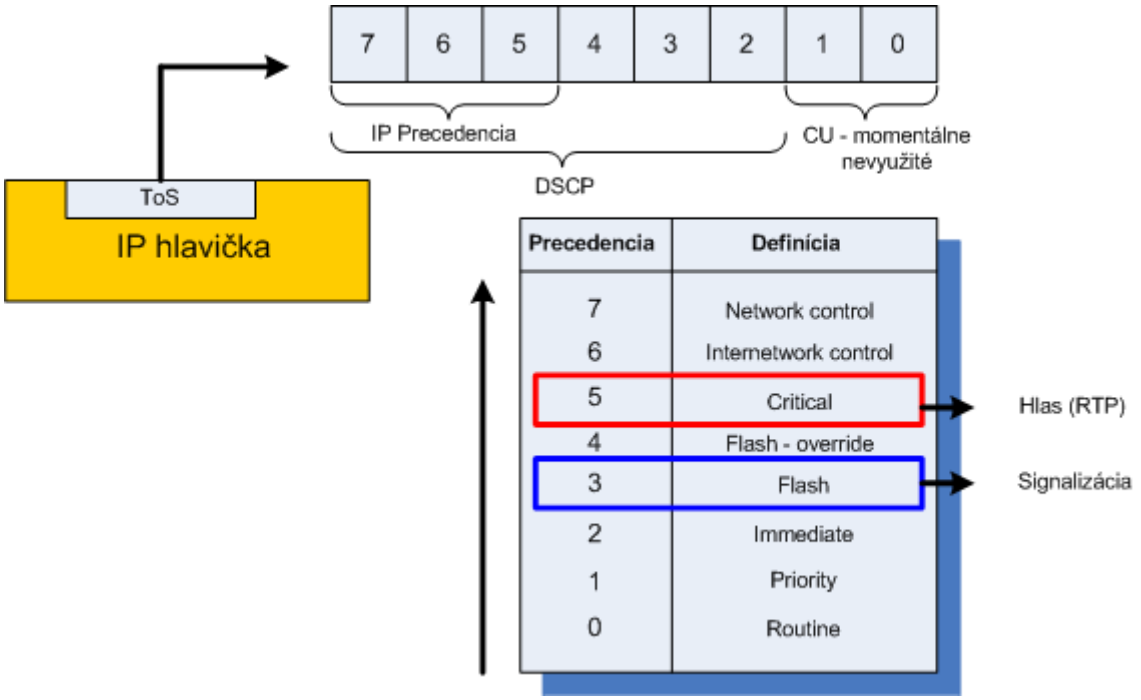
3.5.2 Marking

Vo všeobecnosti sa oddeľujú techniky radenia do frontov (Queuing) od metód značkovania (Marking/Tagging). Vedie k tomu jednoduchá úvaha a to, že queuing poskytuje štruktúru, kým marking indikuje či ten ktorý paket obsahuje hlas, a tým následne poskytne na spracovanie správnemu frontu.

Jeden jednoduchý spôsob značkovania paketov súvisí so základnou charakteristikou prenosu hlasu. V našej testovacej sieti bol hlas jediným typom prenášaných dát využívajúcich UDP/RTP pakety. Pri použití výlučne cisco zariadení začínajú čísla portov pre UDP na 16384 (iný výrobcovia používajú iné rozsahy). Ak by sme použili tento jednoduchý spôsob, mohli by sme na každom sieťovom rozhraní pre radenie do front nastaviť access list aby uprednostňoval UDP pakety s týmito číslami portov. Toto môžeme

dosiahnuť s Priority Queuing (PQ), WFQ, CBWFQ a LLQ (spôsoby radenia do front ktoré sú vysvetlené ďalej v texte).

Iná cesta je použitie IP Precedencie. Je možné nakonfigurovať tzv. dial peer v sieti (zariadenie v sieti ktoré môže iniciovať ako aj prijímať hovory) ktorý by priradil IP Precedence Tag všetkým hlasovým dátam ktoré vygeneruje. Zvyčajne sa používa IP Precedencia rovná 5 na priradenie priority hlasu (rozsah je od 0 do 7, kde 7 má najvyššiu prioritu pre sieťovú prevádzku spojenú s riadením siete). Pre signalizáciu sa používa hodnota rovná 3. Nasledujúca schéma ukazuje ako WFQ automaticky priraďuje takémuto typu dát najvyššiu prioritu bez špeciálnej konfigurácie.



Obr. 3.5.2 – 1 IP Precedence a DSCP bity

Na obrázku vidieť, že pole Type of Service (ToS) je veľké 1 bajt. Tri najvýznamnejšie bity definujú typ IP precedencie a zvyšné bity sú nastavené na 0. Napríklad hodnota IP precedencie typu 5 by mala v ToS poli hodnotu 160.

V súčasnosti sa už používa DSCP značenie používane pri DiffServ QoS modeli. Šesť najdôležitejších bitov ToS bajtu je teda použitých na definovanie DS triedy: DSCP reprezentuje týchto šesť bitov a zvyšné dva bity sú označené ako CU, čiže Currently Unused.

Mapovanie IP Precedence do DSCP vrátane DS triedy je zobrazené v nasledujúcej tabuľke:

IP Precedence	IP Precedence hodnoty bitov	DSCP bity	DSCP Trieda

5	101	101000	Expedited Forwarding
4	100	100000	Assured Forwarding 4
3	11	11000	Assured Forwarding 3
2	10	10000	Assured Forwarding 2
1	1	1000	Assured Forwarding 1
0	0	0	Best Effort

Tab. 3.5.2 – 1 Mapovanie IP Precedence do DSCP

Per hop behavior (PHB) popisuje ako sa daná DS trieda bude správať k paketom v zmysle straty paketov, oneskorenia alebo variabilného oneskorenia. PHB určuje ako sa bude alokovať šírka pásma, ako bude obmedzovaný dátový tok a ako sa budú pakety zahadzovať v prípade zahltenia.

V DiffServ modeli sú definované tri typy PHB:

- **Best-effort trieda** — Default selector bity nastavené na 000
- **Class selelctor** – Mapovanie IP precedence
- **Assured Forwarding** PHB—Class selector bity nastavené na 001, 010, 011, alebo 100
- **Expedited Forwarding** PHB—Class selector bity nastavené na 101

Assured Forwarding (AF) štandard špecifikuje štyri garantované triedy šírky pásiem a popisuje ako sa bude k takémuto typu dát sieť správať. Taktiež definuje tzv. Drop preference úrovne, čo spolu ústi do 12 možných AF tried ako je to znázornené v tabuľke.

Drop Preference úrovne	Trieda AF1	Trieda AF2	Trieda AF3	Trieda AF4
Low Drop Precedence	001010	010010	011010	100010
Medium Drop Precedence	001100	010100	011100	100100
High Drop Precedence	001110	010110	011110	100110

Tab. 3.5.2 – 2 Triedy Assured Forwarding

AF triedy sa zvyčajne aplikujú na sieťovú prevádzku spojenú s prenosom dát, vo väčšine prípadov TCP pakety bez nutnosti prioritného spracovania. EF (Expedited Forwarding) trieda zase viac zodpovedám požiadavkám VoIP QoS.

3.5.3 Queuing

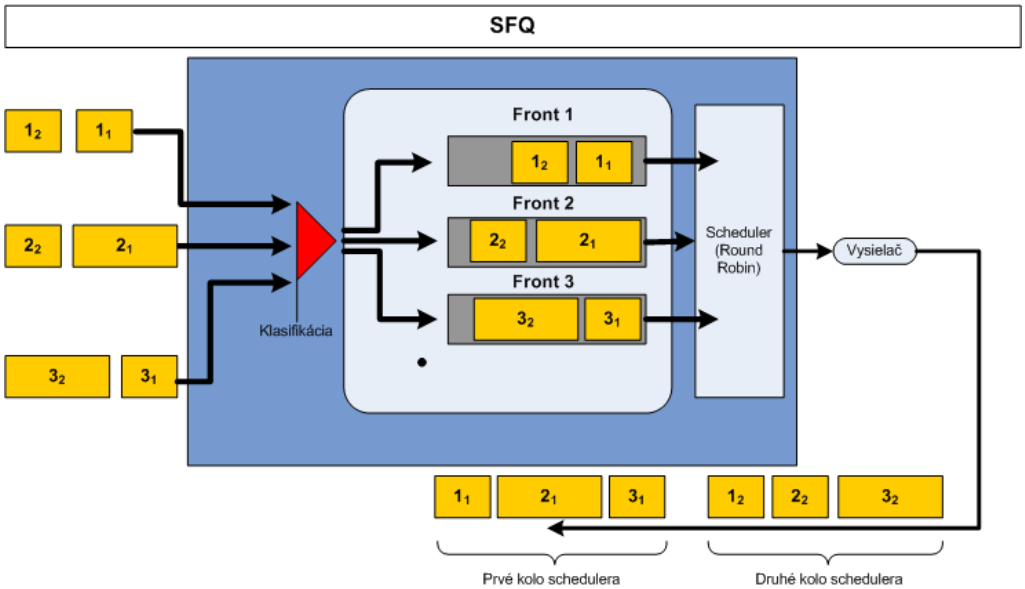
Radenie do frontov ktorá „buffruje“ pakety čakajúce na odoslanie. Je to mechanizmus ktorým paketové siete absorbujú nárazové vlny dát prekračujúce kapacitu linky. Ak pakety dorazia na koniec frontu, tak budú zdržané. Ak sa front naplní, tak sa zahodia pakety najnižšej priority.
Keď sa prenáša hlas cez paketové siete, radenie do frontov dáva vo všeobecnosti prednosť hlasu pred dátovými tokmi. Teda ak by mal byť nejaký typ sieťovej prevádzky príliš zdržiavaný a zahadzovaný, určite by to nemal byť hlas. Teda každá technika radenia do frontov musí byť schopná odlíšiť hlasové pakety od tých dátových a dať takýmto paketom prednosť pri prechode frontami.
V mojich testoch som na použitých zariadeniach mal možnosť overiť podporu rôznych metód radenia do frontov:

- Stochastic Fairness Queuing (SFQ)
- Hierarchy Token Bucket (HTB)
- Weighted Fair Queuing (WFQ)
- Priority Queuing WFQ (PQWFQ) (tiež známe ako IP RTP priority)
- Class-Based WFQ (CBWFQ)
- Low-latency queuing (LLQ) (známe ako Priority Queue, Class-Based Weighted Fair Queuing [PQCBWFQ])

Momentálne najnovšou metódou je naposledy spomenuté LLQ. Táto metóda radenia do frontov postupne vytláča ostatné metódy.

3.5.3.1 Stochastic Fairness Queuing (SFQ)

SFQ patrí k rodine Fair Queuing (FQ) algoritmov a je to systém radenia do front bez použitia tried. Tento konkrétny systém radenia do front nie je dostupný v tejto forme na zariadeniach Cisco aké som použil pri svojich testoch, je k dispozícii vo všetkých novších jadrách operačného systému Linux.
Fair Queuing znamená, že každému dátovému prúdu v sieti ktorý prechádza daným smerovačom, je priradený osobitný front. Klasifikátor takto priradí každému z prúdov nejaký front v závislosti od jeho poradia.



Obr. 3.5.3.1 – 1 Schéma fungovania Stochastic Fairness Queuing (SFQ)

Scheduler v tomto prípade funguje na princípe Round Robin. Ten funguje tak, že v každom cykle vyberie scheduler z každého frontu, jeden po druhom, paket a vloží ho na výstupné rozhranie. Cyklus končí keď sa z každého frontu vyberie jeden paket. Pri FQ je ešte nutné dodať, že jednotlivým prúdom dát je prideľovaná šírka pásma rovnomerne iba v prípade, že je počet paketov teda „dĺžka“ prúdu rovnaká. Ak nie sú rovnaké, tak sa dostupná šírka pásma rozdelí podľa nasledovného vzorca:

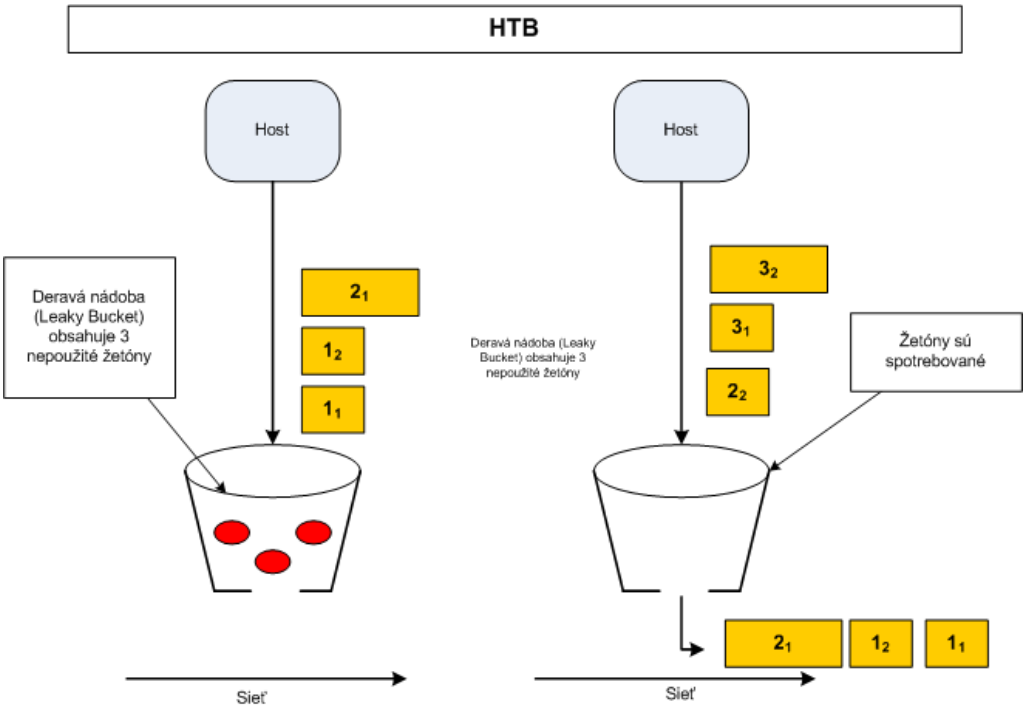
$$B * L_i / (L_1 + \ldots + L_n)$$

B – šírka pásma
Li – stredná dĺžka paketového prúdu

SFQ v podstate spracúva pakety a ich prúdy rovnako ako FQ, avšak disponuje navyše vlastným hešovacím algoritmom, ktorý prúdy dát rozdelí do limitovaného počtu frontov. Hešovacia funkcia sa v určitých okamihoch mení aby sa zabránilo zahlteniu viacerých relácií v tzv. Bucket - nádobe.
SFQ tak ako aj FQ je bez použitia dodatočných mechanizmov pre VoIP nepoužiteľné pretože rozdeľuje dostupnú šírku pásma rovnomerne pre všetky prúdy dát a hlasovým dátam nie je venovaná žiadna dodatočná starostlivosť.

3.5.3.2 Hierarchy Token Bucket (HTB)

HTB je traffic shaping mechanizmus, ktorý funguje na princípe Token-Bucket algoritmu (Token Bucket Filter). Tento algoritmus taktiež pracuje bez využitia tried a funguje tak ako je znázornené na nasledujúcej schéme:



Obr. 3.5.3.2 – 1 Schéma fungovania Hierarchy Token Bucket (HTB)

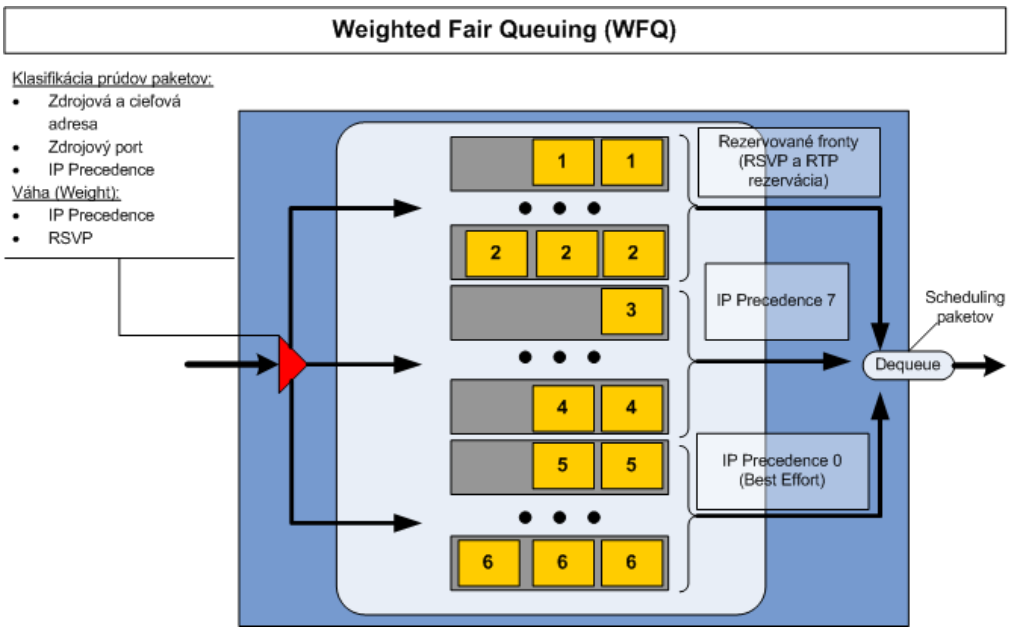
Ako je na obrázku vidieť, vo fronte čaká na prenos sedem paketov a v nádobe (Leaky Bucket) sa nachádzajú tri tokeny (alebo žetóny). Tieto tokeny sa vytvárajú s určitým časovým odstupom na základe taktu časovača. Na to aby sa mohol daný paket preniesť,

musí spotrebovať jeden token. Takýmto spôsobom možno až do nového vytvorenia tokenov preniesť len tri pakety. Cez HTB sa vytvára istá hierarchia v ktorej sa jednotlivé prúdy dát považované za triedy. Vlastne celý tento algoritmus je akousi alternatívou k mechanizmom pracujúcich na základe tried a je veľmi populárny hlavne v open-source svete.

3.5.3.3 Weighted Fair Queuing (WFQ)

WFQ vo svojej základnej verzii rozdeľuje sieťovú prevádzku na rozhraní do prúdov dát, určuje pre každý prúd jeho prenosovú rýchlosť a potom každému prúdu nastaví „váhu“, číže prioritu. Z pohľadu WFQ existujú dve kategórie prúdov dát: High-bandwith relácie a Low-bandwidth relácie. Low-bandwith sieťovej prevádzke sa dáva prednosť pred High-bandwith reláciami a tento typ sieťovej prevádzky si delí zostávajúce sieťové zdroje podľa priradených váh.

WFQ v základnom nastavení je samo objavujúce – to znamená že tento mechanizmus oddeľuje a meria rozdielne prúdy dát a potom im priraďuje váhu. V sieti, ktorá prenáša VoIP sa normálne predpokladá, že hlasové dáta sú low-bandwith prúdy a preto by sa im bude prideľovať vyššia váha, ale toto nie je odporúčaná konfigurácia. Pre silnejšiu prioritizáciu môže byť WFQ nakonfigurované aby dávalo absolútnu prioritu dátam založených na IP precedencii alebo na RTP/UDP čísle portu. Nasledujúca schéma konceptuálne ukazuje ako WFQ spracúva dáta.

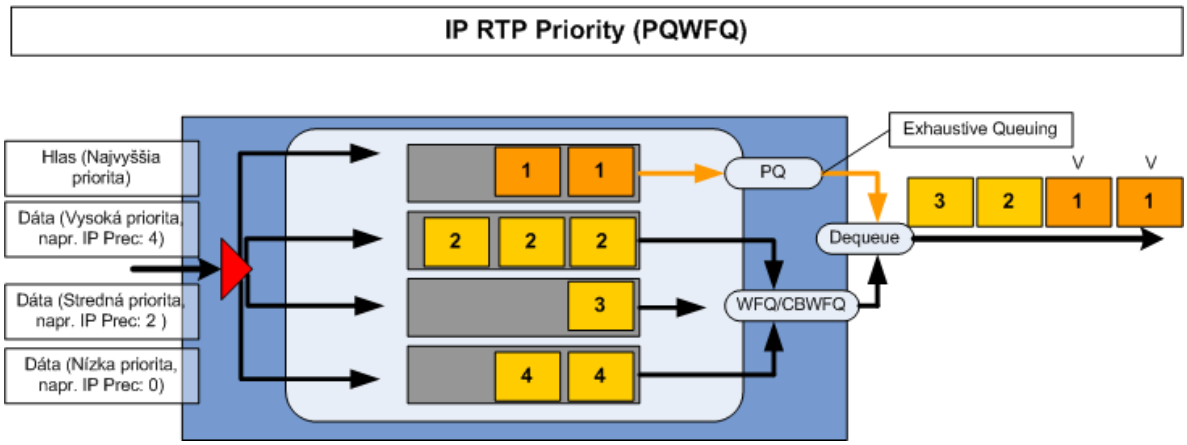


Obr. 3.5.3.3 – 1 Schéma fungovania Weighted Fair Queuing (WFQ)

3.5.3.4 IP RTP Priority

IP RTP priority queuing (tiež známe ako Priority Queue, Weighted Fair Queuing alebo PQWFQ) je schopnosť, ktorá poskytuje striktnú schému prioritného radenia do frontov pre dáta senzitívne na oneskorenie akým je aj hlas. Pri použití PQWFQ sú hlasové dáta identifikované ich číslom RTP portu a sú zaradené do prioritného frontu nakonfigurovanom pomocou príkazu *ip rtp priority*. Toto umožňuje hlasu získať totálnu prednosť v spracúvaní služieb pred nie-hlasovými dátami.

Keď je prioritná fronta prázdna, ostatné fronty sú obsluhované ako štandardné WFQ. Nasledujúca schéma konceptuálne ukazuje ako PQWFQ pracuje.



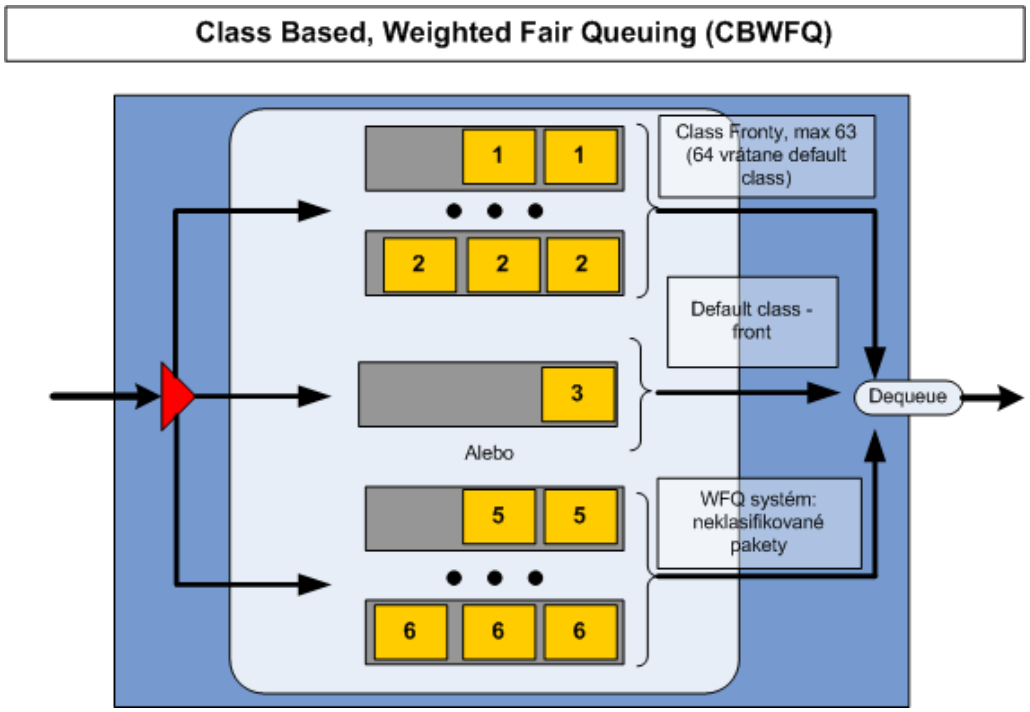
Obr. 3.5.3.4 – 1 Schéma fungovania IP RTP Priority

S PQWFQ si môže administrátor špecifikovať presné množstvo šírky pásma pridelené prioritnej hlasovej prevádzke. PQWFQ podrobne sleduje využitie tejto šírky pásma a zahadzuje hlasové pakety len keď je prekročená. Táto možnosť garantuje hlasovú kvalitu tým, že priraďuje VIP štatút hlasovým dátam.

3.5.3.5 Class-Based Weighted Fair Queuing

CBWFQ dovoľuje definovať špecifickú triedu pre hlasové dáta použitím štandardných a rozšírených číslovaných Access control listov (ACL), protokolov ako aj názvov rozhraní. Pakety ktoré potom spĺňajú kritériá pre danú triedu sa stanú sieťovou prevádzkou danej triedy. To poskytuje trochu hrubšie delenie ako klasifikácia založená na prúdoch.
Každá trieda zadefinovaná pomocou CQWFQ je asociovaná so špecifickým frontom. My si potom môžeme zadefinovať minimálne množstvo garantovanej šírky pásma pre danú triedu a to buď v percentách linky alebo priamo v kbps. Nevyužité prostriedky môžu byť potom zdieľané inými triedami v proporcionálne podľa ich váhy.

Aj keď CBWFQ neprideľuje hlasovým dátam absolútnu prioritu ako je to v prípade PQWFQ, môžeme nastaviť váhy tak aby sa simuloval prioritný front. Pretože je váha pre daný paket patriacemu k nejakej triede odvodená od šírky pásma pridelenej danej triede pri jej vytvorení, je váha v podstate užívateľsky konfigurovateľná. Ďalej nasleduje konceptuálna schéma fungovania CBWFQ.

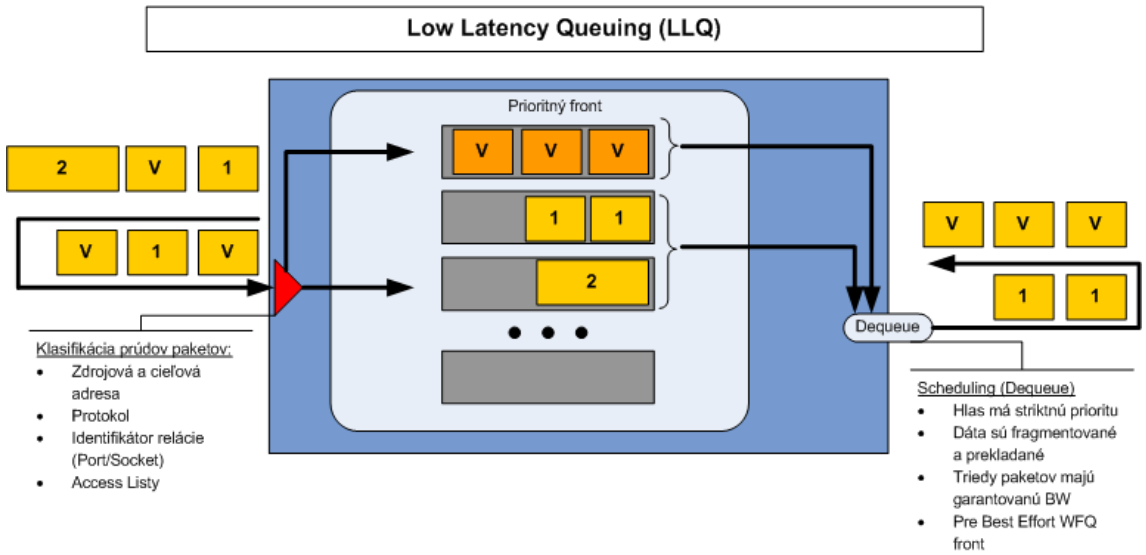


Obr. 3.5.3.5 – 1 Schéma fungovania CBWFQ

3.5.3.6 Low Latency Queuing

Low latency queuing (ďalej už len LLQ), tiež známe ako Priority Queuing, Class-Based Weighted Fair Queuing (PQCBWFQ), je kombinácia predchádzajúcich dvoch metód (PQWFQ a CBWFQ). V tejto metóde je možné triedy hlasu konfigurovať s istou prioritou X namiesto pre istú šírku pásma X, aby sa zabezpečila šírka pásma pre hlas a dala sa mu priorita. Zvyšok sieťovej prevádzky môže byť tiež klasifikovaný a na ten sa použije CBWFQ.

LLQ ponúka také isté zaobchádzanie s hlasovými paketmi ako pri PQWFQ ale ponúka omnoho viacej možností pri konfigurácii a tiež omnoho granulárnejšiu kontrolu nad tým čo môže ísť do prioritného frontu (PQ). LLQ je v funkcionálne nadradené CBWFQ, takže všetko čo môžeme robiť s CBWFQ môžem tiež robiť s LLQ. Nasledujúca schéma ilustruje fungovanie LLQ.



Obr. 3.5.3.6 – 1 Schéma fungovania Low Latency Queuing (LLQ)

3.5.3.7 Weighted Random Early Detection

Random Early Detection (RED) je v skutočnosti metóda pre zabraňovanie zahlcovania vysokorýchlostných TCP sietí ako nejaký mechanizmus pre prioritizáciu hlasových dát. Keď sa niektorý front blíži k zahlteniu, mala by byť použitá metóda Weighted Random Early Detection (WRED) pre zahadzovanie paketov (miera závisí od konfigurácie) z každého vysokorýchlostného dátového prúdu. Toto zabezpečí, že zdroje vysielania sa prepnú do tzv. slow-start módu, čo v konečnom dôsledku zredukuje zahltenie na linke. RED/WRED funguje dobre na dátových tokoch ktoré dobre znášajú zahadzovanie paketov (napr. majú zabudované mechanizmy pre retransmisiu a timeouty). RED a WRED detekcia môže a mala by byť použitá tam kde je prítomný prenos hlasu ale nie na hlas samotný – používa sa na zvyšné typy sieťovej prevádzky (dáta), ktoré zahlcujú rozhrania ktoré sa snaží využiť hlas. Keďže VoIP prenosový protokol je bežne UDP tak WRED ani nemá ako zasiahnuť do tohto prenosu keďže funguje iba pri TCP prenosoch.

RED sa neodporúča používať pri protokoloch ako sú napríklad AppleTalk alebo Novell, ktoré odpovedajú na zahadzovanie paketov retransmisiou rovnakej frekvencie. Odporúča sa ho preto konfigurovať na rozhraniach s prevažne TCP sieťovou prevádzkou.
Pre potreby VoIP sietí nie je WRED prioritizačnou metódou, je skôr zameraná na predchádzanie zahlcovaniu siete. Avšak na širokopásmových chrbticiach veľkých sietí (linky od T3/E3 a vyššie) bude vždy trochu oneskorenia navyše spôsobeného radením do front a práve WRED je vhodnou metódou pre použitie na dátové toky ktoré zdieľajú rozhranie s hlasom.

Poznámka: Kedy nie sú metódy radenia do front v sieti nutné? V podstate queuing je dôležitý iba v sieťach, kde môže nastať nejaká forma zahltenia. Keď máme k dispozícii viac šírky pásma na linke ako dát ktoré po nej prenášame (napríklad také 50% zaťaženie) a zdroje danej linky sú také, že viaceré dátové rámce môžu byť vo fronte pred daným hlasovým rámcom bez zbytočného zdržiavania, fronty budú relatívne prázdne a tieto metódy teda nie sú podstatné. Za týchto okolností môže pre VoIP fungovať aj metóda FCFS – čiže čo prv príde to je prv spracované (jednoduché FIFO).

3.5.4 Traffic Shaping a Policing

Tak shaping ako aj policing mechanizmy sa v sieti používajú na obmedzovanie množstva dát ktoré prúdia do siete a oba mechanizmy používajú klasifikáciu aby mohli dané typy sieťovej prevádzky oddeliť. Taktiež využívajú metering na meranie množstva dát a porovnávajú ho s nakonfigurovanými hodnotami. Rozdiel medzi „shapingom“ a „policingom“ môže byť popísaný rozdielom v ich implementácii obmedzovania množstva dát:

- Shaping meria množstvo dát a zvyšuje oneskorenie paketov ktoré už sú nad rámec nakonfigurovaných hodnôt. Pri použití shapingu sú nárazové vlny dát zmiernené, produkujúc stabilnejší prúd dát. Redukovaním týchto nárazových vĺn sa zamedzí aj zahlcovaniu v jadre siete čo má pozitívny dopad aj na hlasové prenosy.
- Policing všetky pakety ktoré už nie sú v špecifikovanom limite zahadzuje alebo preklasifikuje, napr. na nižšiu prioritu (Best Effort). Policing teda paketom nespôsobuje žiadne oneskorenie. Môže avšak spôsobiť viac TCP retransmisií pretože dáta nad rámec sú zahadzované.

Metódy traffic shapingu a policingu sú aplikované väčšinou pre Frame Relay alebo ATM siete. Jednoducho povedané ak je typ média v sieti leased line (prenajatá linka), tak sa traffic shaping nepoužije. Vo všeobecnosti je dôležité nakonfigurovať traffic shaping daného smerovača tak aby tzv. commited information rate (CIR) Frame Relay siete nebol prekročený. To by mohlo viesť k nadmerným oneskoreniam alebo strate hlasových paketov, z čoho každé by degradovalo kvalitu hlasu. Traffic shaping aj Policing sú veľmi komplexné problémy a to hlavne vo Frame Relay sieťach a jeho úplné vysvetlenie je nad rámec tejto diplomovej práce. Taktiež v testovacej sieti nie je použitý žiadny z týchto mechanizmov.

3.5.5 Fragmentácia

Fragmentácia môže byť nutná ak linka, po ktorej prenášame hlas a dáta, má malú priepustnosť.

Priepustnosť linky (kbps)	56	64	128	256	512	728	1000+
Veľkosť fragmentu (bajty)	70	80	160	320	640	1000	Nie je potrebná*
*Predpokladáme maximálnu veľkosť paketu 1500 B							

Tab. 3.5.5 – 1 Závislosť veľkosti fragmentov od rýchlosti linky

Dáta v predchádzajúcej tabuľke sú všeobecným pravidlom: zámerom fragmentácie je zabezpečiť aby sa hlasový paket príliš nezdržal ak musí čakať na dátový paket ktorý má byť vyslaný na linku pred ním. Je zrejmé, že oneskorenie závisí od šírky dostupného pásma linky a množstve dát v pakete. Vyššie uvedená tabuľka zabezpečí, že oneskorenie z pohľadu hlasového paketu nebude väčšie ako 10 milisekúnd, čo je veľmi konzervatívny postoj.

Na to aby sme zistili hodnotu oneskorenia ktorú si môžeme dovoliť je potrebné:

1. Určiť trasu najhoršieho scenára pre paket v našej sieti (wors-case route).
2. Pridať najhoršie možné oneskorenie v závislosti od čakania vo fronte, oneskorenia vplyvom šírenia a dejitter buffrov.
3. Nakoniec, odčítame túto sumu od maximálneho dovoleného oneskorenia pre hlas v sieti – väčšinou je to od 150 do 200 msec. Výsledok bude určovať oneskorenie spôsobené fragmentáciou ktoré si môžeme dovoliť.

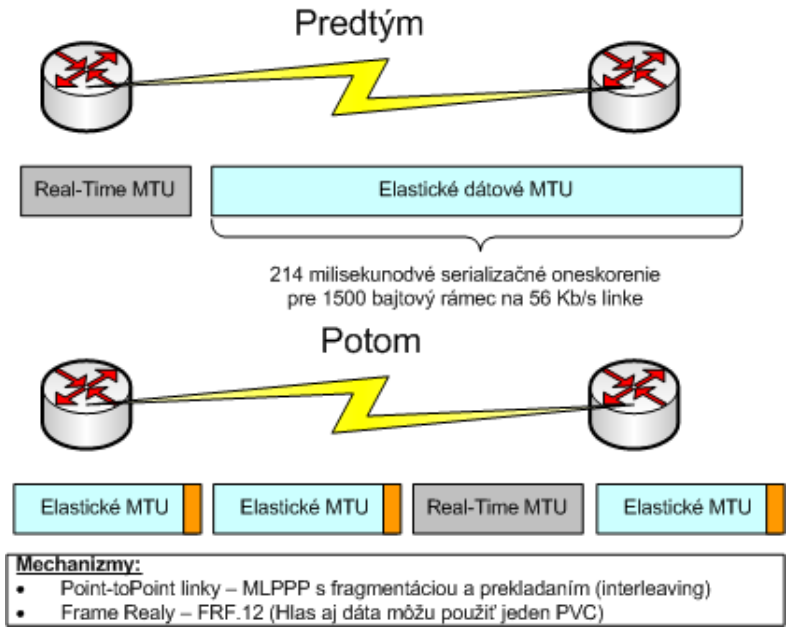
V jednoduchých sieťach sa toto oneskorenie môže pohybovať v rozmedzí od 20 do 50 milisekúnd. V komplikovanejších sieťach to bude bližšie avizovaným 10 milisekundám predpokladaným v uvedenej tabuľke. V rozľahlých sieťach sa určite oplatí vypočítať hodnotu tohto oneskorenia, pretože akékoľvek poľavenie obmedzení na veľkosť fragmentu zlepší celkovú výkonnosť siete (cez nižší packet rate sa redukuje vyťaženie procesorov a tiež sa znižuje množstvo hlavičiek (overhead)).

Takže v sieťach kde sú kapacity liniek malé (menej ako 1 Mb/s), a podľa nariadenia ITU čo sa týka celkového oneskorenia, je fragmentácia na spoločných linkách pre dáta a hlas nevyhnutná. Ak je fragmentácia potrebná, ale nie je použitá, žiadny sofistikovaný mechanizmus radenia do front kvalitu hlasu nenapraví.

Techniky Fragmentácie - Existuje niekoľko fragmentačných metód. Každá metóda má rozdielnu charakteristiku pre použitie v špecifických situáciách.

- Multilink PPP (MLP) s tzv. interleavingom (prekladanie/vrstvenie) je fragmentačná metóda viazaná na každú linku osobitne. To znamená, že veľké pakety sú fregmentované (rozdelené) pre prenos po linke na jednom konci a znova zostavené na prijímacom konci linky – necestujú po celej sieti ako fragmenty. MLP je takisto transparentné pre všetky protokoly (napríklad TCP/Internetwork Packet Exchange – IPX/SNA). MLP je vo všeobecnosti preferovaná metóda fragmentácie na prenajatých linkách, ale v starších IOS ju nebolo možné použiť na Frame Realy linkách (toto obmedzenie už dnes neplatí).
- FRF.12 je štandard Frame Realay fóra a je to považovaný ako alternatíva k MLP. Má všetky výhody ako MLP zahrňujúc:
  - Overhead na line je relatívne nízky
  - Môže byť aplikovaný na každý PVC osobitne
  - Fragmentation Treshold je nastaviteľný
  - Je protokolovo transparentný

  Môže byť síce aplikovaný iba na Frame Relay linky, ale aj prenajaté okruhy môžu byť nakonfigurované aby používali FR enkapsuláciu.
  V prípade, že používame FRF.12 na PVC tak metóda radenia do frontov je automaticky LLQ. Ak z nejakého dôvodu je potreba PQ alebo CQ, ak MLP je lepšia voľba.



Obr. 3.5.5 – 1 Funkcionalita fragmentácie

IP MTU obmedzenie veľkosti - Alternatívna metóda fragmentácie je IP maximum-transmission-unit (MTU) size restriction. IP MTU fragmentuje IP pakety spracované smerovačom ak sú tieto pakety väčšie ako stanovený limit. Fragmenty potom putujú sieťou ku koncovému prijímaču (a nie k poslednému smerovaču), ktorý potom znova zostaví veľký paket. Táto metóda má relatívne veľa nedostatkov: Niektoré IP koncové body nedokážu znovu zostaviť paket z fragmentov menších ako 256 bajtov. A v každom prípade je nutné prekonfigurovať aplikácie ktoré prijímajú fragmentované dáta na koncových zariadeniach.

Použitie IP MTU fragmentácie taktiež vystavuje CPU smerovačov zvýšenému zaťaženiu a to hlavne kvôli zvýšenej miere pps (packets per second) v rámci celej siete. Fragmenty

sa nezostavujú na druhej strane linky ale pokračujú cez celú sieť v takomto stave. IP pakety označené ako „nefragmentovať" budú preposlané ak ich veľkosť nepresiahne nakonfigurovanú hranicu MTU. Ak by boli väčšie tak budú zahodené. Iné (nie IP) protokoly väčšie ako MTU treshold môžu byť prenesené bez fragmentácie, čo by blokovalo hlasové konverzácie, alebo môžu byť zahodené v závislosti na konfigurácii.

3.6 Link efficiency mechanizmy v Cisco IOS

Cisco IOS ako softvér, poháňajúci smerovače od tejto spoločnosti, ponúka päť zefektívňujúcich mechanizmov pracujúcich na linkovej vrstve OSI modelu a to sú: Link Fragmentation and Interleaving (LFI) pre Multilink PPP (MLP), LFI pre Frame Realy a ATM virtuálne okruhy, Frame Relay Fragmetácia, Compressed Real-Time Protocol (CRTP) a distribuovaný CRTP (dCRTP). Všetky tieto mechanizmy spolupracujú s metódami ako queuing a traffic shaping aby sa zabezpečila efektívnosť a predvídateľnosť služieb aplikačnej vrstvy.

Interaktívne dáta (akými sú aj VoIP pakety) sú náchylné na zvýšené oneskorenie ak sieť zároveň prenáša aj veľké dátové pakety. Oneskorenie hlasových paketov je badateľné hlavne ak je pred našimi hlasovými paketmi vo výstupnej fronte veľa veľkých paketov a linka je pomalá. Pre nízko-rýchlostné linky (menej ako 1Mb/s) je potrebná nejaká metóda, ktorá by rozkúskovala veľké dátové pakety a tým dovolila malým hlasovým paketom aby boli vložené medzi tieto fragmenty (interleaving) a takto poslané ďalšiemu uzlu v sieti.

Taktiež je nutné komprimovať relatívne veľké RTP hlavičky aby sa zvýšila dostupná priepustnosť linky. Na tento účel sa preto používa komprimovaná hlavička v podobe cRTP.

Termín *link efficiency mechanisms* teda popisuje softvérové metódy a príkazy ktoré túto fragmentáciu a prekladanie robia na sériových ako aj frame-relay linkách a taktiež komprimujú hlavičky paketov. Dosahuje sa tak vyššia efektivita a predvídateľnosť aplikácií pracujúcich na týchto linkách.

V mojich praktických testoch sa teda budem zameriavať na aplikáciu LEM a to:

- Link fragmentation a interleaving (LFI) pre multilink point-to-point protokol (MLP) sériové linky
- Kompresiu hlavičiek RTP protokolu – cRTP

Od týchto metód očakávam zníženie zaťaženia nízko rýchlostných ppp spojov a tiež zníženie oneskorenia a jittru VoIP paketov.

Ďalšie optimalizácie budem vykonávať nastavovaním rôznych priorít pre hlasové pakety v rôznych frontoch na výstupných rozhraniach smerovačov. Metódy ktoré zahadzujú pakety prípadne ich oneskorujú ako sú shaping, policing prípadne WRED pri výskyte zahltenia nemá pre VoIP zmysel nasadzovať – kvalita hlasu sa tým môže výrazne zhoršiť. Takéto mechanizmy sa nasadzujú hlavne pre sieťovú prevádzku spojenú s nárazovými vysokými požiadavkami na sieťové zdroje ako sú napr. P2P služby, VoD alebo zálohy serverov a iné spojovo orientované TCP prenosy. V mojej konfigurácii použijem WRED na rozhraniach cez ktoré bude prúdiť generovaný tok balast dát.

3.7 QoS a VoIP signalizácia

Zabezpečenie signalizácie VoIP prípadne akejkoľvek inej signalizácie v sieti by malo byť rovnako dôležité ako zabezpečenie dát ktoré prenášajú konkrétny hlas. V prípade že by boli signalizačné pakety z nejakého dôvodu zahadzované alebo oneskorované v dôsledku zahltenia buffrov na niektorom rozhraní smerovača po ceste, spomalilo by to celkovo nadviazanie a ukončenie hovoru, čo by malo za následok degradáciu vnemu kvality služby.
Je preto vhodné pre takýto typ dát vytvoriť vlastný typ fronty na smerovačoch a teda vlastnú triedu s vlastným DSCP/IP precedence kódom. V súčasnosti sa najčastejšie používajú tieto typy VoIP signalizácie v sieťach.

Protokol	**Transport & Port**
Skinny (Cisco telefóny a Call Manager)	TCP 2000 – 2002
H.323	TCP 1720, 11XXX
MGCP	UDP 2427, 2428
SIP	TCP, UDP 5060

Tab. 3.7 – 1 Signalizačné protokoly a ich porty

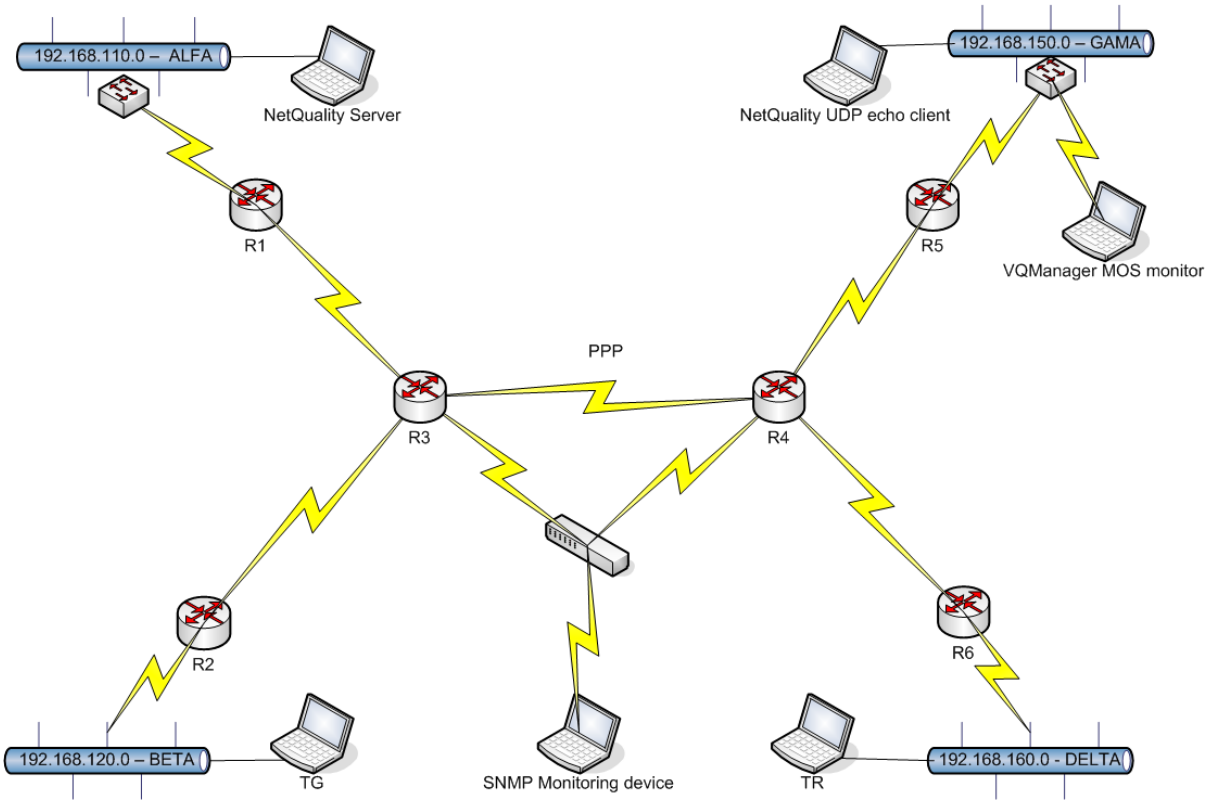
V mojom riešení navrhujem použiť pre tento typ sieťovej prevádzky nastavenie ToS poľa v paketoch na hodnotu DSCP AF31 prípadne ekvivalentu IP Precedence level 3. Pre tieto malé pakety vyskytujúce sa najčastejšie pri zahajovaní a ukončovaní hovorov je to postačujúce riešenie.

PRAKTICKÁ ČASŤ

4. Použité zariadenia a softvérové prostriedky

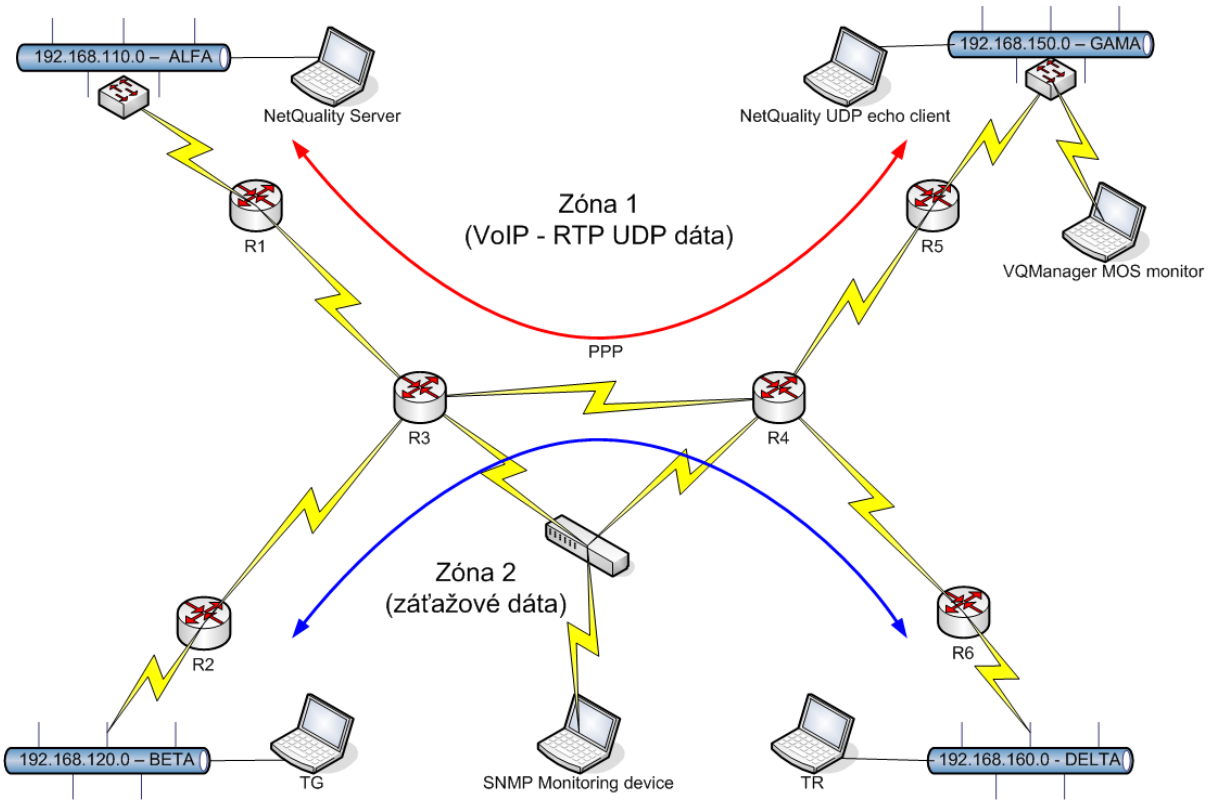
4.1 Model siete

Pre testovacie účely som v laboratóriu KIS B301 vytvoril sieť na ktorej bolo možné overiť funkčnosť popísaných mechanizmov a navrhnúť optimálne riešenie pre siete s podobnou konfiguráciou. Sieť pozostáva zo smerovačov a prepínačov značky Cisco.



Obr. 4.1 – 1 Model testovacej siete

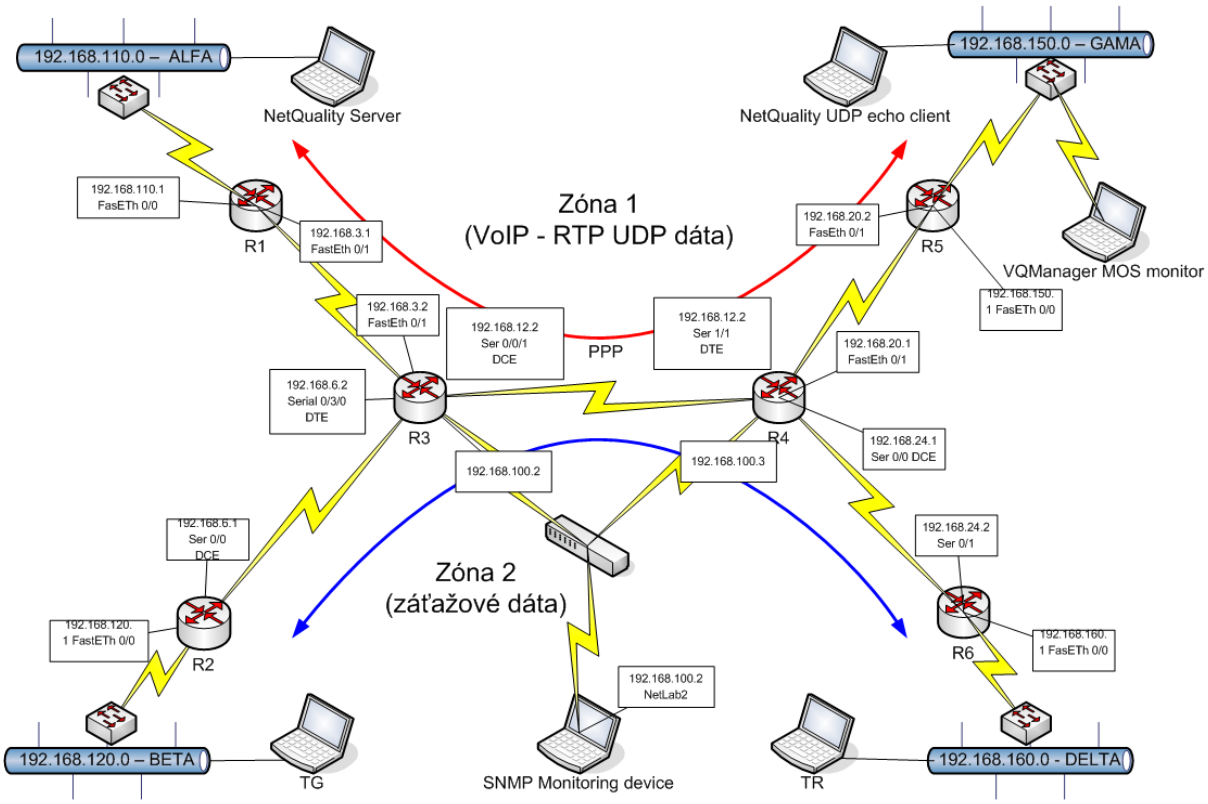
Sieť pozostáva zo štyroch subsietí typu LAN a ďalších ppp spojov medzi jednotlivými smerovačmi. WAN spojenie s PPP enkaspuláciou je simulované medzi smerovačmi R3 a R4. Na tejto linke bude dochádzať k zahlcovaniu a stretu generovaných dát z TG (Traffic Generator) a dát vygenerovaných pomocou testovacieho programu simulujúcom RTP UDP sieťovú prevádzku, prípadne s reálnymi VoIP dátami z telefónov. Sieť som v rámci zabezpečenia oddelenia jednotlivých druhov sieťovej prevádzky rozdelil na dve zóny tak ako je to znázornené na nasledujúcom obrázku.



Obr. 4.1 – 2 Záťažové zóny v testovacej sieti

Týmto sa zabezpečí, že sa dáta stretnú až na WAN spoji kde je pre obmedzenú šírku pásma nutné použiť všetky dostupné metódy pre zabezpečenie hlasu. V testovacích scenároch budem postupne meniť šírku pásma a konfiguráciu práve medzi smerovačmi R3 a R4. Takáto konfigurácia môže simulovať prípad keď sú dve budovy od seba vzdialené a prepojené sú pomocou prenajatých okruhov (leased lines) prípadne iným typom nízkorýchlostného spoja (=< 2 Mb/s). V konfigurácii budem predpokladať použitie QoS metód na smerovačoch R3 a R4 a použitie klasifikácie a značkovania na smerovačoch R1 a R5. R1 teda bude predstavovať ingress smerovač a R3 zase egress smerovač, analogicky v sieti na druhej strane WAN spoja.

Podrobná konfigurácia siete aj s rozhraniami je znázornená na nasledujúcej schéme. (Schéma sa nachádza v prílohe 9.5)



Obr. 4.1 – 3 Podrobná schéma testovacej siete

Každej stanici na koncových subsieťach je pridelená IP adresa pomocou DHCP z daného adresného priestoru.

4.2 Použité zariadenia

Na vytvorenie prostredia na ktorom by bolo možné simuláciu a testovanie siete vykonať som použili zariadenia dostupné v laboratóriu počítačových sietí CCNP. Všetky použité zariadenia boli vyrobené spoločnosťou Cisco Systems. Zariadenia rovnakého typu sa od seba odlišovali iba použitými modulmi.

Cisco 1841 – typ smerovačov použitý v topológii ako R1, R2, R5 a R6 uzly. Použitá verzia IOS (Internetworking Operating System) bola na všetkých smerovačoch rovnaká na to aby sa vylúčili chyby spôsobené možnou nekompatibilitou a hlavne dostupnými funkciami.
Tento model je náhradou série 1700.

Cisco 2811 – Tento ty smerovača som použil pre spoj medzi R3 a R4 smerovačmi v testovacej sieti. Smerovač využíva 256 MB internej RAM pamäte a 64 MB pamäte flash.

Obr. 4.2 – 1 Cisco 2811

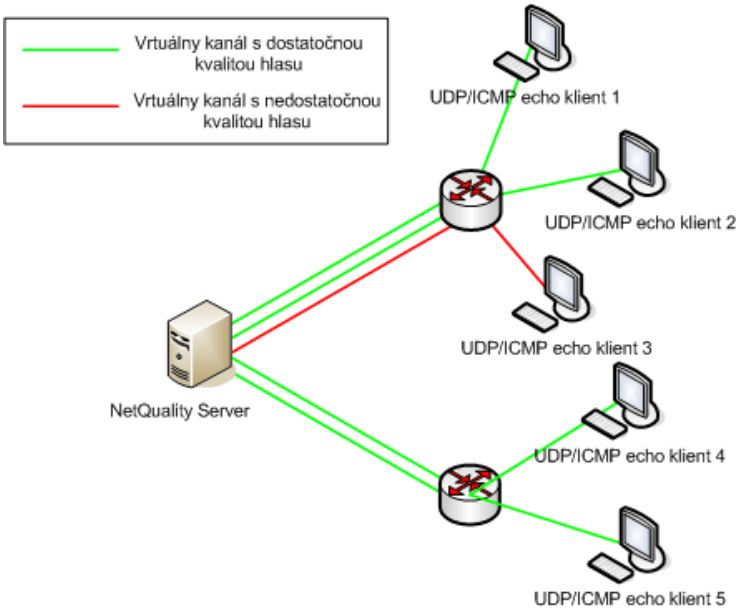
Ako prepínače boli použité Cisco Catalyst série 2900 na ktoré boli priamo zapojené koncové stanice s meracími nástrojmi. Samotné pracovné stanice boli potom PC s konfiguráciami uvedenými v tabuľke.

NetQuality Server	AMD Sempron 3300+ 1GB RAM	MS Windows XP Professional SP2
SNMP Monitor	1GHz AMD Duron 512 MB RAM	MS Wndows 2000
VQManager MOS monitor	AMD Sempron 3300+ 1GB RAM	MS Windows XP Professional SP2

V každom z konkrétnych testov bolo výhodné použiť reálne IP telefóny pre fyzické overenie kvality prenášaného hlasu a tak potvrdiť alebo vyvrátiť výsledky z nameraných hodnôt. V testoch boli použité dva typy IP telefónov a to softvérové a hardvérové. Ako softvérový telefón bol zvolený X-lite, odľahčená freeware verzia Eye-Beam VoIP telefónu od firmy Counter Path. Zástupcom hardvérového IP telefónu bol Telco PH-800N.

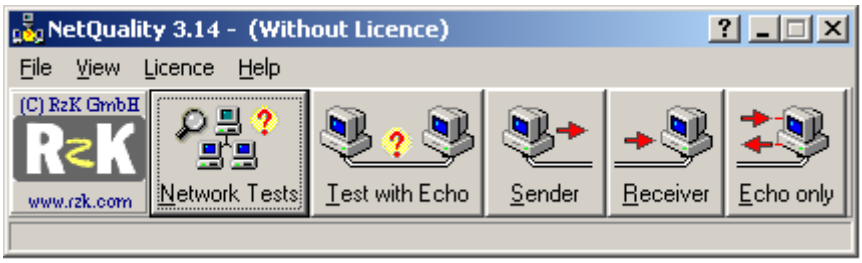
4.3 Použité meracie prostriedky

Aby som mohol vyhodnotiť parametre ako jitter, oneskorenie alebo stratu paketov bolo zapotreby meracieho nástroja, ktorý by dokázal nasimulovať VoIP hovory. Po dlhšom hľadaní som sa nakoniec rozhodol použiť softvér NetQuality VoIP od nemeckej firmy RzK (www.rzk.com). NetQuality VoIP je merací systém (tzv. assessment systém) pre vyhodnotenie VoIP možností IP sietí momentálne dostupný iba pre platformu MS Windows. Pomocou NetQuality simulovaných hovorov sa dajú merať jednotlivé QoS parametre a taktiež graficky vyhodnotiť. NetQuality VoIP sa inštaluje na vybranom uzle v sieti, ktorý bude slúžiť ako server na testovanie linky. Ostatné stanice v sieti potom slúžia ako klienti tak ako je to znázornené na obr. 4.3. – 1. Server vysiela prúdy RTP dát ku klientom na koncových staniciach a tie tieto dáta zrkadlia späť na server tak, aby bola zabezpečená obojstranná sieťová prevádzka. Po prijatí týchto dát dokáže server vyhodnotiť parametre jednotlivých prúdov.



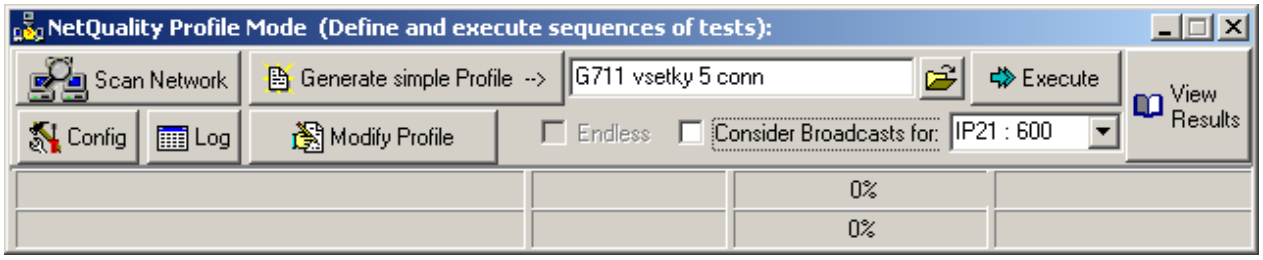
Obr. 4.3 – 1 Schematické zapojenie NetQuality

NetQuality VoIP Server - Po štarte softvéru NetQuality sa objaví hlavné menu programu:



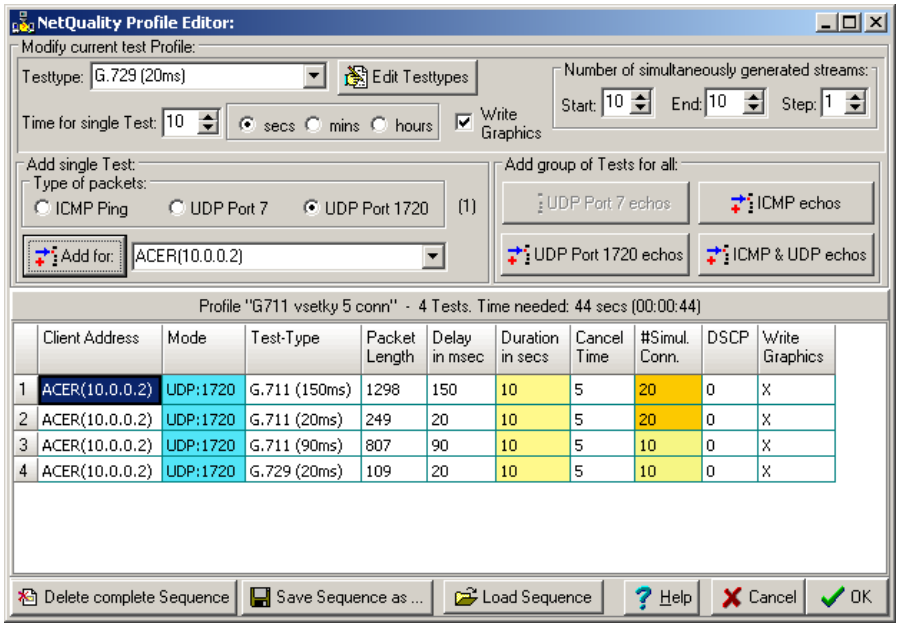
Obr. 4.3 – 2 Hlavné menu NetQuality

V hlavnom menu sa dá zvoliť spomedzi rôznych typov testov, či už jednotlivých alebo komlexných sieťových meraní. Pretože potrebujeme vykonať merania s rôznymi parametrami a pri rôznych podmienkach zvolíme z hlavého menu položku „Network Tests“. Tu je možné detailne nakonfigurovať všetky potrebné parametre.



Obr. 4.3 – 3 Okno spúšťania sieťových testov

S použitím submenu „Scan Network“ sa najskôr vykoná prehľadanie danej siete alebo definovanej podsiete na definované UDP echo alebo ICMP odpovede, týmto sa zisťuje na akých staniciach v sieti sa môžu vykonať aké série testov.
V menu „Create Detailed Profile“ alebo „Modify Profile“ sa potom vytvoria jednotlivé testovacie profily.



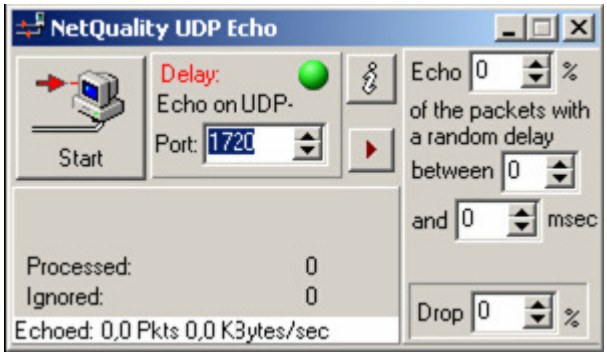
Obr. 4.3 – 4 Vytváranie sieťového testu

V každom vytvorenom profile sa dá navoliť až 1000 možných echo testov, ktoré majú tieto parametre:

- Client Address: IP alebo MAC adresa, poprípade NetBIOS alebo DNS názov daného echo klienta.
- Mode: Typ testu – MAC vrstva, ICMP alebo UDP (vrátane portu).
- Test Type: aký kodek bude použitý pre simuláciu (G.711, G.729)
- Packet Length: Veľkosť generovaných paketov.
- Delay in msec: Oneskorenie v milisekundách medzi dvoma paketmi.
- Duration in secs: Celková dĺžka trvania testu.
- Cancel Time: Čas, po ktorom sa test preruší, ak klient nepošle späť žiadne pakety.
- #Simul. Conn: Number of simulated connections – počet zároveň generovaných prúdov VoIP dát.
- Write Graphics: či sa majú zhotoviť grafické zobrazenia jednotlivých parametrov v priebehu testu. (delay, jitter, loss).

Po vytvorení požadovaného testovacieho profilu sa tento spustí cez tlačidlo „Execute“.

NetQuality UDP echo client - Pre testy pomocou UDP echo ktoré som použil vo svojich testoch aj ja, má NetQuality VoIP k dispozícii jednoduchý program s názvom UDP – Echo Client, ktorý je nainštalovaný na koncových staniciach. Program je taktiež možné stiahnuť z IP adresy bežiaceho NetQuality Servera, keďže ten disponuje vlastným web serverom.



Obr. 4.3 – 5 NetQuality UDP echo klient

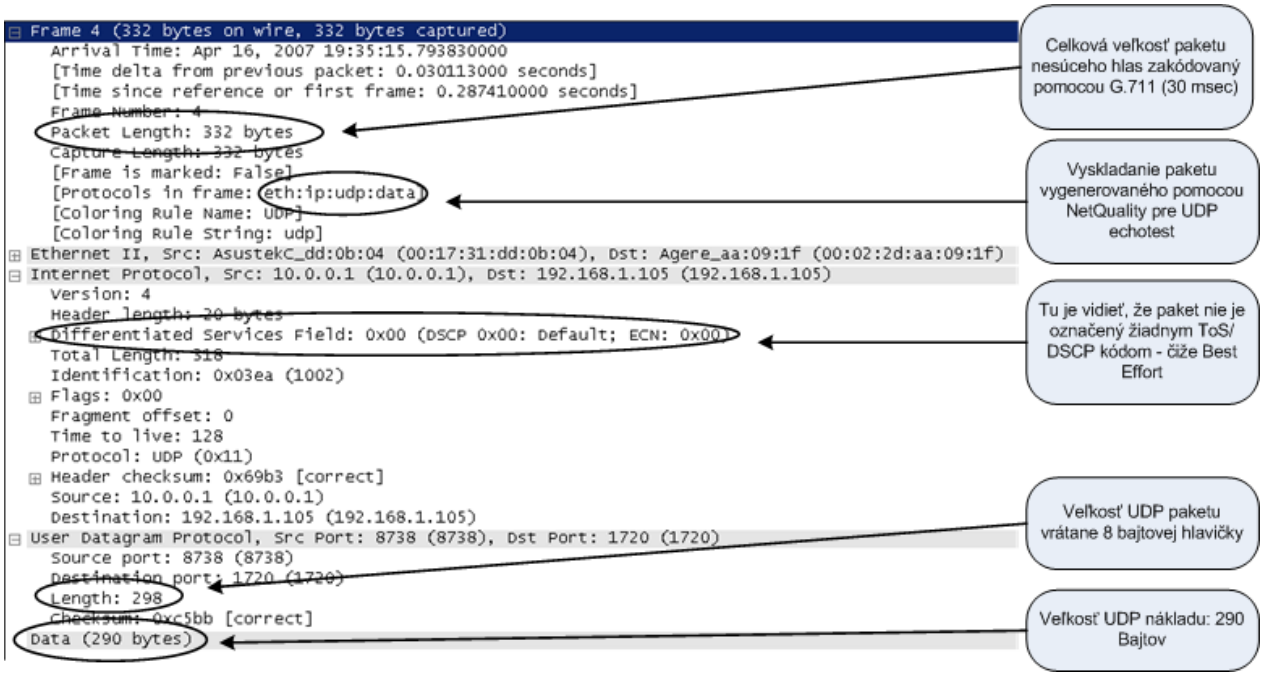
NetQuality Server simuluje jednu stranu virtuálneho kanálu medzi dvoma telefónmi kým Echo klient simuluje stranu druhú. Spúšťa sa tlačidlom „Start“ a má nasledovné parametre:

- Echo on UDP port: UDP číslo portu pre prichádzajúce pakety, musí byť zhodné s číslom portu v testovacom profile.
- Echo: Percento paketov, ktoré budú odoslané späť serveru s nejakým náhodným oneskorením (v milisekundách).
- Drop: Percento paketov, ktoré budú na strane klienta zahodené.

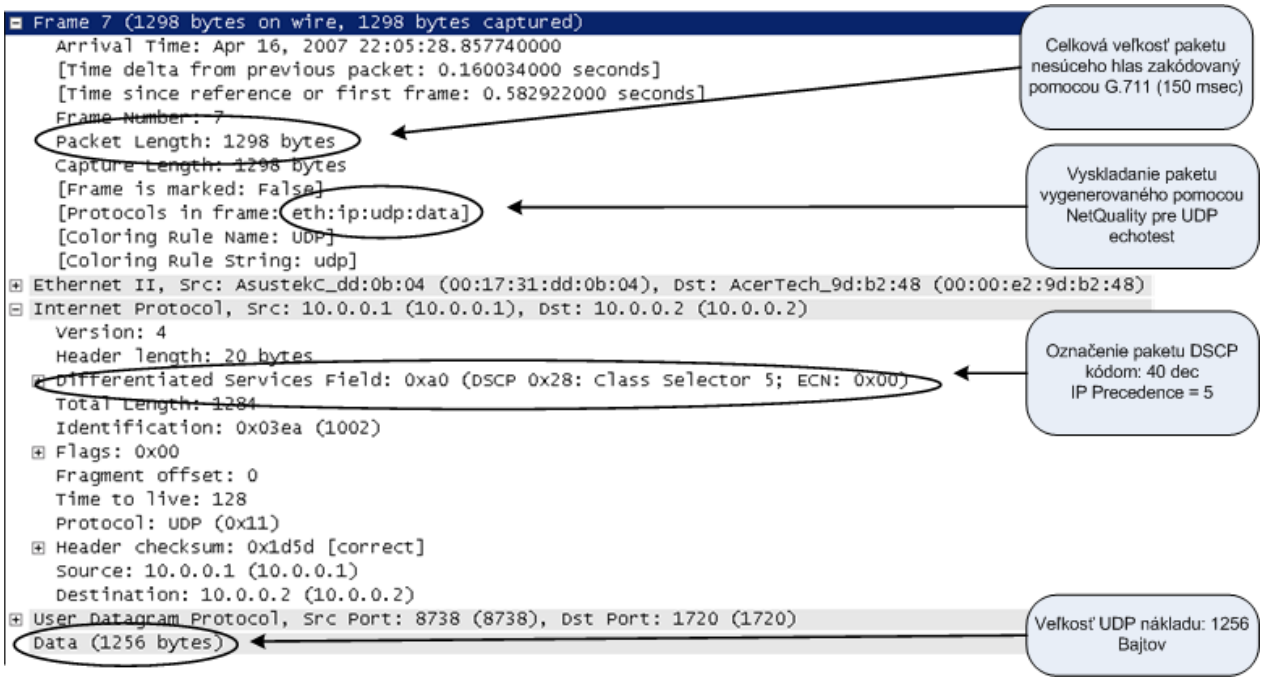
Nastavenia pre oneskorenie a stratu paketov boli v mojich testoch vynechané, ich hodnoty boli nastavené na 0.

Princíp fungovania - NetQuality server vždy odosiela svoje generované pakety počas testov z UDP portu 8738 a taktiež ich na tomto porte aj prijíma. Cieľový port je nastaviteľný ale UDP echo klienti majú štandardne port nastavený na UDP 1720. Nasledujúci obrázok ukazuje štruktúru paketu získaného cez sieťový „sniffer“ Wireshark. Jedná sa o rámec G.711 paketu s 30 milisekundovým oneskorením, ktorý bol vygenerovaný pomocou NetQuality Servera (zdrojový port je 8738). Ako je vidieť, VoIP

hovory sú simulované cez UDP pakety so zodpovedajúcim dátovým obsahom, ktorého veľkosť sa líši v závislosti od zvoleného kodeku a oneskorenia.

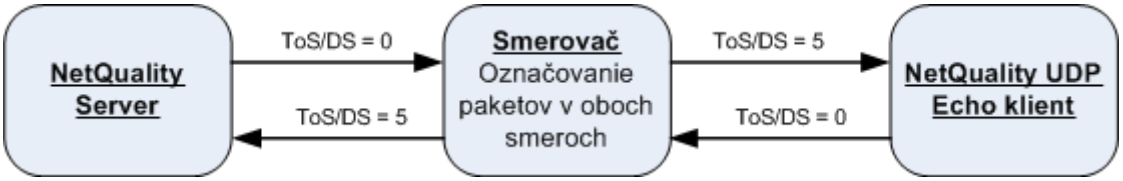


Jednotlivé kodeky a ich štandardné oneskorenia teda získame porovnaním veľkosti nákladu jednotlivých paketov.
Pre porovnanie, nasledujúci obrázok je taktiež paket získaný prostredníctvom Wiresharku a ukazuje paket vygenerovaný s kodekom G.711 a 150 milisekundovým oneskorením. Paket bol dodatočne označený DSCP kódom.



NetQuality poskytuje možnosť označenia paketov DSCP kódom už pri jeho generovaní, avšak akonáhle paket dorazí ku klientovi, ten paket vracia už bez označenia. Preto som marking presunul na smerovače R1 a R5 kde bola paketom vsúvaná ich priorita.

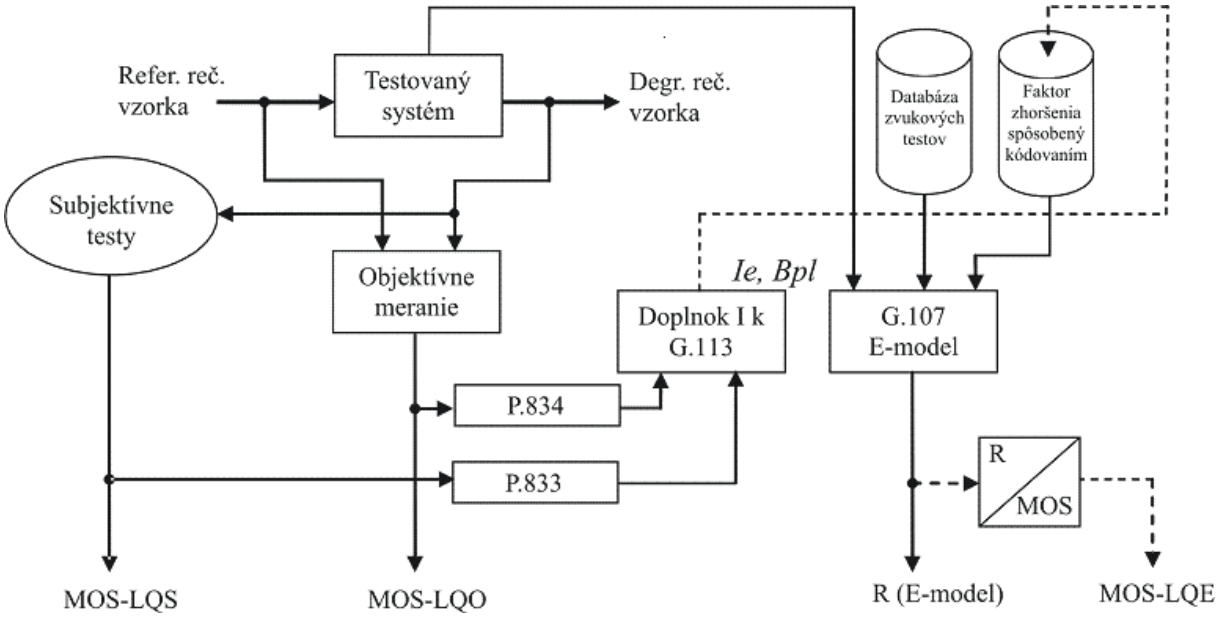
V princípe označovanie paketov fungovalo ako je znázornené na nasledujúcej schéme:



Obr. 4.3 – 6 Marking paketov v testovacej sieti

ManageEngine™ VQManager - ManageEngine™ VQManager je webovo orientovaný real-time QoS monitorovací nástroj pre VoIP siete. VQManager môže monitorovať hocijaké zariadenie alebo user agenta ktorý podporuje SIP (RFC 3261) a RTP/RTCP (RFC 3550). Aplikácia beží na serveri Tomcat a je naprogramovaná v Jave a Java Server Pages. V podstate sa jedná o serverovskú časť aplikácie ktorá beží ako služba na pozadí a vizuálne rozhranie ktoré spravuje tomcat.

VQManager vyhodnocuje MOS pasívne tým že počíta R-faktor ktorý môže byť použitý na odhadnutie MOS. Nasledujúca bloková schéma určuje vzájomný vzťah medzi jednotlivými typmi meraní. Pričom doporučenie ITU-T Rec. P.834 určuje metodológiu získania faktora zhoršenia z objektívnych meraní a doporučenie ITU-T Rec. P.833 zo subjektívnych meraní. Ďalej doplnok I k doporučeniu ITU-T Rec. G.113 z roku 2002 eviduje databázu faktorov zhoršenia a odolnosti kodeku voči strate paketov pre rôzne typy kodekov. Prevod prenosového činiteľa R na hodnotu MOS je určený prevodovou krivkou určenou experimentálne.



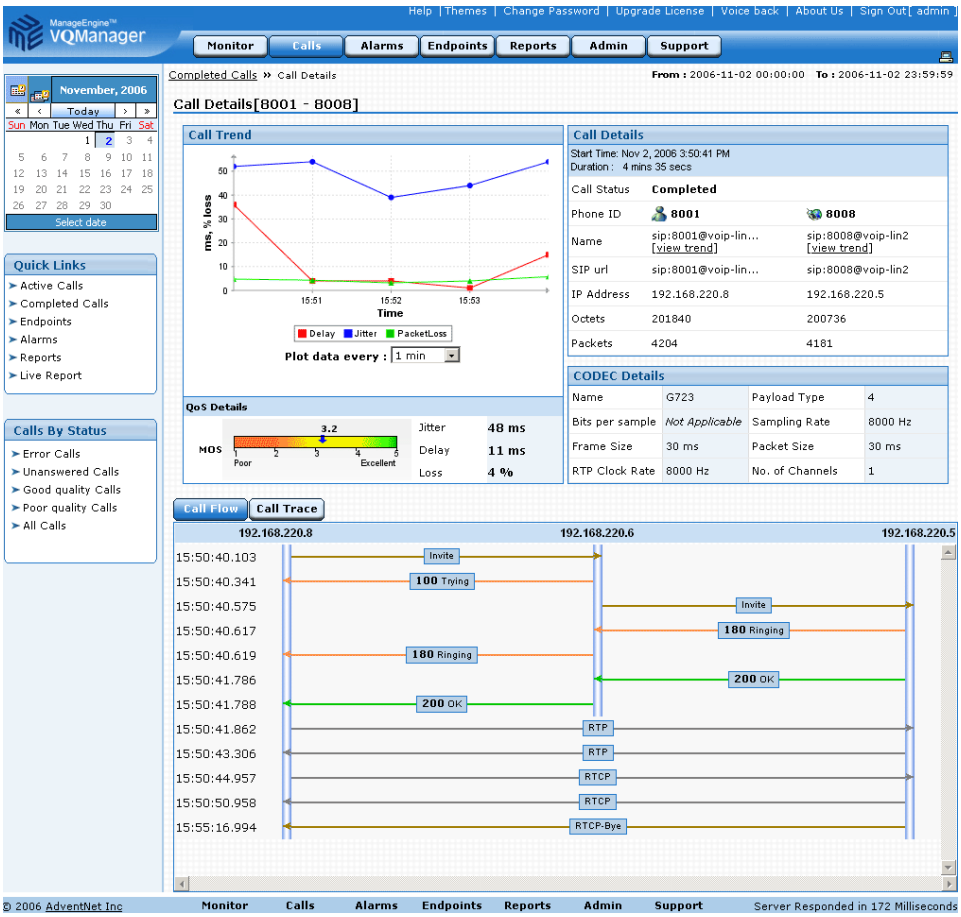
Obr. 4.3 – 7 Podrobná schéma vyhodnocovania MOS

V podstate práca VQManagera spočíva v tom, že sleduje sieťovú prevádzku na sieti a podľa SIP správ a RTP prúdov dát alebo RTCP paketov odhaľuje a monitoruje prebiehajúce hovory na sieti. V prípade, že koncové zariadenia nevysielajú RTCP pakety je nutné toto nastaviť v nastaveniach vo web rozhraní.
Na to aby sme mohli sledovať a vyhodnocovať dáta na sieti, bolo nutné aby boli meracie nástroje vsunuté do siete. Lenže transparentné sniffovanie (dátový tok by „pretekal“ zariadením) by nebolo možné kvôli možnému zvýšeniu oneskorenia čo by podstatne skreslilo namerané hodnoty. Preto bolo nutné spraviť „mirroring“ všetkých dát, teda naklonovanie trafficu. Toto je možné realizovať buď na smerovačoch v jadre siete a to

pomocou vytvorenia GRE tunelu, alebo v prístupovej sieti na prepínačoch. Ja som zvolili druhú variantu, a na prepínačoch Cisco 2950 sa SPAN (port mirroring) konfiguroval týmto spôsobom:

```
C2950#configure terminal
C2950(config)#
C2950(config)#monitor session 1 source interface fastethernet 0/2 both

!--- Tento príkaz nakonfiguruje rozhranie Fast Ethernet 0/2 ako zdrojový
port z ktorého budem monitorovat pakety.

C2950(config)#monitor session 1 destination interface fastethernet 0/3

!--- Tento príkaz nakonfiguruje rozhranie Fast Ethernet 0/3 ako cieľový
port kde bude počúvať v promiskuitnom režime VQManager.

C2950(config)#
C2950#show monitor session 1
Session 1
---------
Source Ports:
  RX Only:     None
  TX Only:     None
  Both:      Fa0/2
Destination Ports: Fa0/3
```

V použitej konfigurácii boli všetky používané porty zrkadlené na port 24 kde bol zapojený VQManager. Sledoval sa vstupný aj výstupný dátový tok. Nasledujúci screenshot zobrazuje rozhranie VQManagera pri detaile pre jeden dátový tok (hovor).



Obr. 4.3 – 8 Detail parametrov hovoru v rozhraní VQManagera

Tento nástroj mi slúžil hlavne na overenie nameraných hodnôt pomocou NetQuality a taktiež na meranie parametrov hovorov vykonaných cez reálne VoIP telefóny.

PRTG - Paessler Router Traffic Grapher - PRTG je monitorovací nástroj ktorý som použil pre monitorovanie využitia sieťovej prevádzky na linke medzi smerovačmi R3 a R4. Použitý protokol bol SNMP a konfigurácia prebiehala na smerovači R3. V testovacej sieti slúžil hlavne na monitorovanie záťaže a generovaného „overheadu“. Ovládanie a vizualizácia bola riešená cez web rozhranie na monitorovacej stanici kam boli zo smerovače zasielané stavové dáta.

D-ITG – Distributed Internet Traffic Generator - D-ITG som využil pre generovanie záťaže pre sieť. Jeho hlavnou výhodou je jeho konfigurovateľnosť a tak som mohol dynamicky ovládať veľkosti generovaných paketov. Funguje na platforme Windows aj Linux a podporuje IPv4 aj IPv6 protokol. Ako generované typy protokolov sú k dispozícii protokoly UDP, TCP, ICMP, DNS, Telnet alebo aj RTP. Časový interval medzi paketmi ako aj veľkosť paketu závisí od náhodných rozdelení premenných z ktorých máme k dispozícii niekoľko (constant, exponential, uniform, cauchy, paretovo, poiss, gamma). D-ITG taktiež poskytuje možnosť štatistického merania rôznych sieťových parametrov čo však vo veľkej miere závisí na časovej synchronizácii medzi vysielačom a prijímačom paketov. Pokiaľ chceme korektne merať tzv. One Way Delay (OWD) čiže jednosmerné oneskorenie, hodiny prijímača a vysielača musia byť synchronizované (napríklad pomocou NTP alebo GPS). Distribuovaný v tomto význame znamená, že prijímač aj vysielač umožňuje prijímať aj odosielať toky dát paralelne (v rôznych vláknach).

Použitie:

1. zapnúť prijímač(B):

./ITGRecv

2. zapnúť vysielač(A):

./ITGSend -a B -sp 9400 -rp 9500 -O 100 -e 500 -t 20000 -x recv log file

Vlastnosti toku: zdrojový port = 9400, cieľový port = 9500, priemerne je poslaných 100

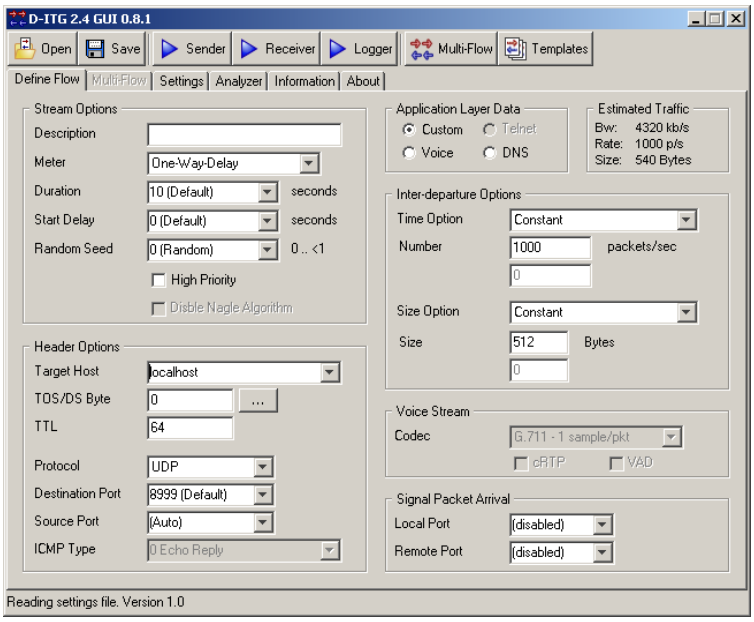
paketov za sekundu, veľkosť každého paketu je v priemere 500 B (exponenciálne

rozdelenie), trvanie testovania je 20 sekúnd (20000 milisekúnd)

3. dekódovať log súbor na strane prijímača

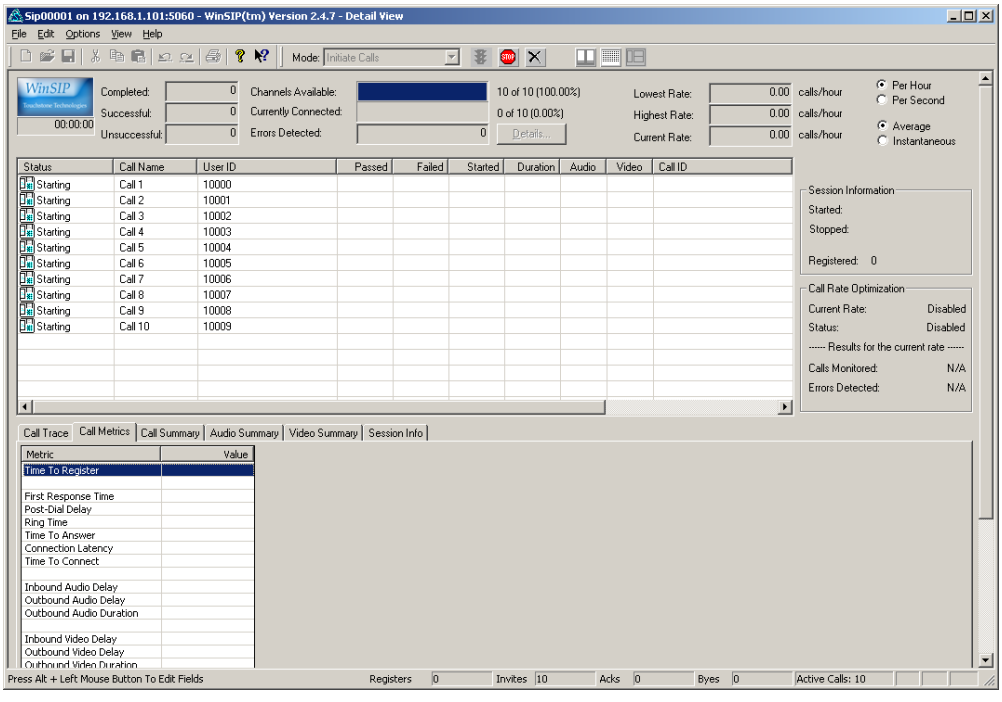
./ITGDec recv log file

Namiesto použitia príkazového riadku je možné použiť grafickú nadstavbu v Jave. Veľkou nevýhodou bola nestabilita programu po niekoľkých spusteniach. Program pred samotným generovaním paketov vytvorí signalizačný kanál kde sa prenášajú riadiace informácie. Tento kanál nebolo možné často vytvoriť a tým nebolo možné test vykonať až do reštartu stanice. Vzhľadom na nezosynchronizované koncové body pri generovaní záťaže nebolo možné korektne vyhodnocovať parametre generovaného toku akou sú stratovosť alebo oneskorenie. Tieto hodnoty v testoch neuvádzam.



Obr. 4.3 – 9 Java rozhranie pre generátor paketov D-ITG

WinSIP - je vysokovýkonný, škálovateľný softvér pre generovanie SIP hovorov. Poskytuje RFC 3261 signalizáciu a taktiež je ním možné generovať aj prislúchajúci RTP stream a to ako hlasový tak aj audiovizuálny a tiež DTMF signály. Je plne konfigurovateľný a umožňuje vytvoriť presne ten typ scenára ktorý zodpovedá testu prípadne reálnej situácii. V mojich testoch som ho použil na meranie parametrov SIP signalizácie a to na zostavenie spojenia a jeho zrušenie a čas na nadviazanie spojenia. Výsledkom testov sa dá určiť ako by zaťažená sieť zniesla ďalší hovor (z pohľadu signalizácie) a či by doba pre nadviazanie spojenia prípadne jeho ukončenia nebol príliš veľká, čo by znížilo celkový vnem z kvality hovoru. V testoch som periodicky generoval jednotlivé hovory popri oddelene generovaným RTP dátam tak aby to zodpovedalo počtu dátových prúdov.



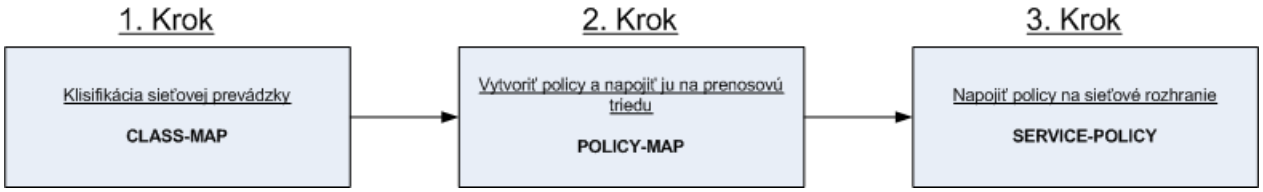
Obr. 4.3 – 10 Rozhranie programu WinSIP

5. Konfigurácia QoS mechanizmov v IOS

5.1 Všeobecná konfigurácia Cisco IOS

Smerovače Cisco v laboratóriu sieťových technológií katedry informačných sietí sú poháňané ich vlastným operačným systémom *Internetworking Operating System* (IOS). Na smerovačoch ktoré som použil pri svojom testovaní boli nainštalované verzie IOS 12.3T a 12.4 ktoré plne podporujú QoS funkcie potrebné pre testovacie scenáre.

Na implementáciu Quality of Service služieb na smerovačoch cisco je potrebné vykonať nasledovné kroky.



Obr. 5.1 – 1 Postup pri konfigurácii QoS na Cisco zariadeniach

Obrázok 5.1 - 1 ilustruje ako vo všeobecnosti nastaviť nejaký typ QoS zabezpečenia na smerovači pomocou CLI (Command Line Interface) rozhrania.
Ako prvý krok v konfigurácii QoS je klasifikácia dát prichádzajúcich na rozhranie smerovača. Na vytvorenie nejake prenosovej triedy je zapotreby vytvoriť „class-map“. Meno tejto class-map je voľne zvoliteľné a malo by súvisieť s jej neskoršou funkciou.
Pomocou príkazu:

„configure terminal“

Sa dostaneme do konfiguračného módu smerovača: (príklad pre hlasové dáta)

```
R3#conf
Configuring from terminal, memory, or netwok [terminal]?
Enter configuration commands, one per line. End with
CNTL/Z.
R3(config)#class-map voice
R3(config-cmap)#
```

Ako ďalší krok je vytvorenie pravidiel v ktorých sa určí, ktoré QoS mechanizmy budú na danú prenosovú triedu použité. Na to sa vytvoria tzv. „Traffic Policies“ takže pravidlá prenosu a na ne sa napojí daná prenosová trieda. Tu platí to isté ako pri vytváraní prenosovej triedy: názov by mal slúžiť pre lepšiu orientáciu (príklad pre odchádzajúce VoIP dáta):

```
R3(config)#policy-map voice_out
R3(config-pmap)#
```

V poslednom kroku konfigurácie sa vytvorené pravidlá prenosu aplikujú na konkrétne rozhranie. Tu je nutné zadať smer pre ktorý majú dané pravidlá platiť, teda či sa jedná o vstupný alebo výstupný smer. (Príklad pre výstupný smer z ethernet 0 rozhrania)

```
R3(config)#interface ethernet 0
R3(config-if)#service-policy output voice_out
```

Na to aby sme mohli IP pakety prichádzajúce zo vstupných rozhraní klasifikovať nám slúžia tzv. Access Listy (ACL). Pomocou ACL môžem na rozhraní nadefinovať, ktoré pakety sa budú ďalej spracovávať a ktoré budú systémom blokované. Jednotlivé ACL sú od seba odlišované pomocou čísla a sú typizované podľa protokolu ako je znázornené v nasledujúcej tabuľke. Zobrazuje len najdôležitejšie protokoly a ich ACL čísla.

ACL Typ	**Rozsah ACL čísiel**
IP ACL	1-99, 1300-1999
Extended IP ACL (napr. VoIP)	100-199, 2000-2699
Ethernet Type Code ACL	200-299
Ethernet Address ACL	700-799

Tab. 5.1 – 1 Typy ACL

Access Listy ponúkajú užívateľovi široké možnosti diferencovania paketových prúdov. IP paketom môžeme taktiež priradiť vyššiu prioritu podľa cieľovej alebo zdrojovej IP adresy alebo podľa určitého cieľového alebo zdrojového portu.
Príklad:

```
R3#conf
Configuring from terminal, memory, or netwok [terminal]?
Enter configuration commands, one per line. End with
CNTL/Z.
R3(config)#access-list 101 permit udp any any eq 1720
R3(config)#
```

V tomto príklade bol vytvorený ACL 101, ktorý povolí všetky pakety so zdrojovým alebo cieľovým UDP portom 1720 (H.323 signalizácia).
ACL ponúkajú pri konfigurácii QoS mechanizmov väčšiu flexibilitu a preto ich používam aj vo svojej konfigurácii.

Klasifikácia paketov v testovacej sieti prebiehala na smerovačoch R1 a R5. Uvediem príklad konfigurácie smerovača R1:

```
access-list 102 permit udp any any range 8700 32767
access-list 103 permit udp any any range 5060 5063
...
class-map match-all signaling-marking
 match access-group 103
class-map match-all voice-marking
 match access-group 102
...
policy-map setdscp
 class voice-marking
  set dscp ef
 class signaling-marking
  set dscp cs3
...
interface FastEthernet0/0
 description "eth subnet 110.0"
 ip address 192.168.110.1 255.255.255.0
 service-policy input setdscp
 duplex auto
 speed auto
```

5.2 Inštalačné skripty

Pri implementovaní QoS mechanizmov vzhľadom na to, že v laboratóriu prebieha rada ďalších testov, nebolo možné uložiť testovacie konfigurácie natrvalo na smerovače. Na to aby sa daná konfigurácia zapísala do NVRAM je potrebné vykonať príkaz:

„copy system:running-config nvram:startup-config“

Pri implementácii som už mal predpripravené konfiguračné skripty pre jednotlivé testy a nahrával som ich cez CLI smerovačov. Po reštarte sa zase nahrala pôvodná konfigurácia z NVRAM.

5.2.1 Konfigurácia LLQ a IP RTP Priority

Na začiatok by som zhrnul vlastnosti týchto metód, ich výhody a nevýhody. Implementované budú na smerovačoch R3 a R4 v oboch smeroch.

Low Latency Queuing (LLQ)	**IP RTP Priority**
Diferencovanie dát podľa: • Access Listy (pre rozsah UDP portov, adresy staníc, IP ToS: IP Precedence, DSCP kód) • IP RTP rozsah portov • IP ToS: DSCP/IP Precedence • Protokol a vstupné rozhranie • Všetky kritéria používane v CBWFQ	Diferencovanie dát podľa: • Založené na RTP/UDP rozsahu portov: (Pre telefóny Cisco 16384 - 32767)
Výhody: • Väčšia flexibilita ako sú pakety vkladané do PQ a CBWFQ • Je možné nakonfigurovať dodatočné triedy aby sa garantovala šírka pásma aj dátam spojeným s VoIP signalizáciou a videom	Výhody: • Jednoduchá konfigurácia
Nevýhody: • Komplexná konfigurácia	Nevýhody: • RTCP pakety ako aj iná signalizácia sú vkladané do bežného WFQ frontu a nie PQ. • Slúži pre VoIP pakety ale postráda konfiguráciu pre ostatné typy prúdov dát

Poznámka: Ak sa plánuje prioritizácia dát v závislosti na iných kritériách ako rozsah UDP portov (napríklad Diffserv PHB) je nutné použiť LLQ.

V nasledujúcom texte je popísaný postup konfigurácie jednotlivých metód zabezpečenia kvality hlasu a jednotlivé kroky sú popísané avšak kompletné konfigurácie sú až v prílohe.

5.2.2 LLQ

Konfigurácia LLQ pozostáva z nasledujúcich krokov:

1. Vytvorenie Class-map pre VoIP pakety a zadefinovanie rozlišovacích kritérií (Konfigurácia je názorná a nemusí sa úplne zhodovať s použitou v smerovačoch)

```
R4(config)#class-map match-all voice-traffic
```

Použijeme access-group pre klasifikáciu dát – využíva access listy

```
R4(config-cmap)#match access-group 102
```

Teraz sa vytvorí access-list na ktorý sa naviaže naša class-map access-group

```
R4(config)#access-list 102 permit udp any any eq 8738
```

Najjednoduchšie je diferencovať na základe UDP portu, ktorý je v našom prípade port NetQuality servera 8738.

Taktiež je možné použiť access- listy ktoré s použitím access-group diferencujú pakety na základe iných parametrov:

```
access-list 102 permit udp any any precedence critical
```

- Tento ACL filtruje pakety na základe IP ToS hodnoty – Precedencie. Je dôležité aby s takouto hodnotou boli označené iba VoIP pakety.

```
access-list 102 permit udp any any dscp ef
```

- Tu ACL pozerá na hodnotu DSCP kódu – v tomto prípade ef (expedited forwarding)

```
Access-list 102 permit udp host 192.10.1.1 host 192.20.1.1
```

- Takýto typ ACL sa používa v prípade, že pakety ktoré pochádzajú od smerovača alebo od koncového zariadenia, nie sú označené pomocou IP Precedence alebo DSCP a nemožno ani určiť rozsah VoIP UDP portov.

V ostatných prípadoch je možné diferencovať pakety aj bez použitia access-groups:

```
class-map voice
     match ip rtp 16384 16383
```

- V tomto prípade je použitie zhodné s IP RTP Priority keď sa pozeráme na UDP porty paketov. Táto funkcionalita je v Cisco IOS od verzie 12.1.2 T.

- Nasledovné dve metódy majú za predpoklad, že pakety prichádzajúce na vstupné rozhranie sú už označené koncovým zariadením alebo smerovačom predtým než sa aplikuje LLQ.

```
class-map voice
     match ip precedence 5
```

alebo

```
class-map voice
     match ip dscp ef
```

2. Vytvorenie Class-map pre VoIP signalizáciu a zadefinovanie rozlišovacích kritérií

```
class-map voice-signaling
match access-group 103

access-list 103 permit udp any eq 5060 any
access-list 103 permit udp any any eq 5060
```

VoIP hovory používajú rôzne signalizačné metódy a preto uvádzam aj alternatívne protokoly a ich porty:

- H.323/H.225 = TCP 1720
- H.323/H.245 = TCP 11xxx (Standard Connect)
- H.323/H.245 = TCP 1720 (Fast Connect)
- H.323/H.225 RAS = TCP 1719
- Skinny = TCP 2000-2002 (CM Encore)
- ICCP = TCP 8001-8002 (CM Encore)
- MGCP = UDP 2427, TCP 2428 (CM Encore)
- SIP= UDP 5060, TCP 5060 (konfigurovateľné)

3. Vytvorenie policy-map a asociácia s VoIP class-map

```
R4(config)#policy-map VOICE-POLICY
R4(config-pmap)#class voice-traffic
R4(config-pmap-c)#priority ?
<8-2000000> Kilo Bits per second
```

Konfigurácia voice-traffic triedy do striktného prioritného frontu a priradenie šírky pásma.

```
R4(config-pmap)#class voice-signaling
R4(config-pmap-c)#bandwidth 8
```

Priradenie 8 Kb/s triede voice-signaling

```
R4(config-pmap)#class class-default
R4(config-pmap-c)#fair-queue
```

Zvyšné pakety sú spracované vo WFQ.

Poznámka: Aj keď je možné radiť do prioritnej fronty prakticky akékoľvek pakety, odporúča sa výhradne pre hlasové pakety. Real-time dáta ako video by mohli vyvolať zvýšenie variabilného oneskorenia načo sú hlasové pakety omnoho senzitívnejšie ako video. Taktiež platí, že suma hodnôt *priority* a *bandwith* by nemala prekročiť 75% kapacity linky. Zvyšok kapacity by mal byť vyhradený pre smerovacie protokoly a taktiež by priamo nebolo možné uplatniť service-policy (bola by vypísaná chybová správa).

4. Povolenie LLQ – Aplikácia policy-map na výstupné rozhranie

```
R4(config)#interface multilink 1
R4(config-if)#service-policy output VOICE-POLICY
```

V tomto scenáriu (MLPPP LFI) je service-policy naviazaná na Multilink rozhranie.

5.2.3 IP RTP Priority

Na konfiguráciu IP RTP Priority je na výstupnom rozhraní potrebné nakonfigurovať:

```
Router(config-if)#ip rtp priority starting-rtp-port port-range bandwidth
```

Starting-rtp-port je najnižší UDP port na ktorý sa pakety posielajú. Port-range je rozsah UDP portov od najnižšieho definovaného portu a bandwidth je je maximálna povolená šírka pásma pre dáta v PQ. Nastavuje sa podľa odhadovaného počtu zároveň prebiehajúcich hovorov v sieti.

Príklad konfigurácie:

```
interface Multilink1
{…}
bandwidth 64
ip address 172.22.130.2 255.255.255.252
ip tcp header-compression
fair-queue
no cdp enable
ppp multilink
ppp multilink fragment-delay 10
ppp multilink interleave
multilink-group 1
ip rtp header-compression iphc-format
ip rtp priority 16384 16383 45
```

5.2.4 Link Fragmentation a Interleaving : Multilink PPP

Kým štandardný dátový paket má veľkosť okolo 1500 bajtov, VoIP paket má bežne veľkosť pohybujúcu sa okolo 66 bajtov (použitý kodek je G.729). Hlavne na pomalých linkách je problém serializácie najväčší keďže oneskorenie môže ľahko presiahnuť hodnoty vyše 200 msec. Veľké pakety preto môžu veľmi nepriaznivo pôsobiť na prenos malých, časovo senzitívnych paketov cez sieť. Fragmentácia týchto paketov a prekladanie týchto fragmentov malými VoIP paketmi výrazne znižuje oneskorenie a jitter. Vo všeobecnosti sa odporúča aby oneskorenie na základe serializácie na jednom uzle bolo okolo 10 prípadne najviac 20 milisekúnd. Fragmentácia je nastaviteľná pomocou príkazu **ppp multilink fragment-delay** a určuje sa v milisekundách. LFI vyžaduje aby bol taktiež povolené prekladanie paketov pomocou **ppp multilink interleave**. Pri linkách s rýchlosťami nad 1Mb/s nie je nutná táto konfigurácia, keďže oneskorenie serializáciou je minimálne.

5.2.5 Compressed Real-time Protocol (cRTP)

cRTP nie je prostriedok QoS ale slúži na zmenšenie veľkosti RTP paketov, čím znižuje zaťaženie linky. Založené na RFC 2508, cRTP komprimuje IP/UDP/RTP hlavičku z pôvodných 40 Bajtov na 2 alebo 4 Bajty. Táto kompresia je typu hop-by-hop a je nutné ju konfigurovať na každom uzle samostatne. Konfigurácia je jednoduchá:

```
Router(config-if)#ip rtp header-compression [passive]
```

Keďže táto kompresia je náročná na CPU, odporúča sa ju používať iba na linkách s kapacitou nižšou ako 1 Mb/s prípadne na linkách kde vyťaženie CPU smerovača nepresiahne 75%. Pri konfigurácii cRTP IOS automaticky pridá do konfiguračného súboru aj kompresiu TCP/IP hlavičiek ako popísané v RFC 1144:

```
Router(config-if)#ip tcp header-compression [passive]
```

Znižovanie zaťaženia siete je možné takisto použitím iného typu kodeku (pri mojich testoch G.711 a G.729) a taktiež použitím VAD (Voice Activity Detection), kedy je možné ušetriť až 35% prenosovej kapacity potlačením prenosu ticha cez sieť.

5.2.6 Auto QoS

Cisco ponúka ako súčasť ich IOS možnosť automatickej konfigurácie QoS s názvom Auto QoS. Jeho použitie veľmi zefektívňuje aplikáciu QoS mechanizmov v sieti a používa všetky už spomenuté metódy. Výhoda je že administrátor nemusí mať dokonalé znalosti týchto mechanizmov a šetrí čas v rozľahlých sieťach.

Príklad použitia:

```
interface Serial0
bandwidth 256
Ip address 10.1.61.1 255.255.255.0
Autoqos voip
```

6. Testovacie scenáre a výsledky testov

Pri návrhu testovacích scenárov som sa snažil nastaviť parametre siete a generovaných tokov tak, aby bolo jasne vidieť rozdiely prípadne nedostatky v prenose hlasu. Najprv som spravil referenčné testy bez použitia QoS metód aby bolo možné porovnanie s výsledkami kde boli použité jednotlivé popísané mechanizmy. Potom nasledovali testovacie scenáre s QoS. V rámci každého scenáru som chcel simulovať dva typy linky – vysokorýchlostnú WAN taktovanú na 2Mb/s a nízkorýchlostnú variantu o rýchlosti 384 Kb/s. Výstupom každého scenára je tabuľka s nasledovnými hodnotami:

SIP time to connect – táto hodnota je výstupom z programu WinSIP a sleduje čas za ktorý sú klienti schopní posielať RTP dáta (INVITE, ACK) .

SIP tear down – taktiež hodnota získaná z programu WinSIP, sleduje čas korektného ukončenia relácie pomocou správy BYE.

Strata rámcov – počet stratených rámcov v rámci prenosu

Strata (Loss) v % - percentuálne vyjadrenie straty paketov

Pseudo strata – je to strata paketov ktorá nastane až na strane volajúceho a je spôsobená použitým kodekom. Táto strata nastáva pri veľkej hodnote variabilného oneskorenia kedy pakety dorazia mimo časový rámec akceptovateľný kodekom.

Pseudo strata v % - percentuálne vyjadrenie pseudo straty.

Priemerné oneskorenie – priemerné oneskorenie paketov v milisekundách

Maximálne oneskorenie – maximálne namerané oneskorenie v milisekundách

Priemerný jitter – priemerná hodnota jittru (vraibilného oneskorenia) v milisekundách

Maximálny jitter – maximálna nameraná hodnota jittru v milisekundách

R–faktor – hodnota R-faktoru vypočítaná z hodnôt VoIP hovoru uskutočnenom z hardvérových telefónov. Tento hovor bol vytvorený pri každom teste z daných scenárov a prenášal dáta pomocou G.711 kodeku. V rámci VQManagera bolo možné merať MOS a R-faktor iba za použitia tohto hovoru keďže generovaný RTP stream programu NetQuality nevysielal RTCP pakety ktoré sú potrebné pre výpočet.

MOS – hodnota mean opinion score uskutočneného hovoru

Každý z pripravených scenárov pozostáva zo série tých istých testov ako znázorňuje tabuľka 6 – 1 a 6 - 2. Počet generovaných hovorv pozostáva z generovaných RTP dát a reálneho hovoru čo je v tabuľke označené ako x+1.

Hlasové vyťaženie siete v %	Použitý kodek	Záťaž v % (Background)	Počet generovaných hovorov	Záťaž na pozadí prenosu hlasu
60%	G.711	30%	2+1	114 Kb/s
90%	G.711	30%	3+1	114 Kb/s
60%	G.729	30%	5+1	114 Kb/s
60%	G.729	100%	5+1	384 Kb/s
60%	G.711	100%	2+1	384 Kb/s

Tab. 6 – 1 Testy pre rýchlosť linky 384 Kb/s

Hlasové vyťaženie siete v %	Použitý kodek	Záťaž v % (Background)	Počet generovaných hovorov	Záťaž na pozadí prenosu hlasu
60%	G.711	30%	13+1	600 Kb/s
90%	G.711	30%	20+1	600 Kb/s
60%	G.729	30%	37+1	600 Kb/s
60%	G.729	100%	37+1	2 Mb/s
60%	G.711	100%	13+1	2 Mb/s

Tab. 6 – 2 Testy pre rýchlosť linky 2 Mb/s

Ako je z tabuliek vidieť, nevykonával som test pri 90 % hlasovom zaťažení kodekom G.729 pretože pri tak vysokom počte hovorov začínala mať stanica na ktorej bežal NetQuality Server značné problémy s výkonom (AMD Athlon 3300 +). Tento test bol preto vynechaný. Taktiež testujem kvalitu prenosu hlasu pri 100 % záťaži na pozadí. Tento test je veľmi náročný na spracovanie smerovačmi a vstupné fronty sú ľahko preplnené keďže teoretická záťaž linky je niekedy až 190 % jej kapacity. Takáto situácia samozrejme v sieti nastať môže avšak nie počas doby akú bol test vykonávaný. Takýto prípad môže nastať pri nárazových vlnách v sieti (tzv. Bursts) a trvá len malú chvíľu. Bolo však určite nutné otestovať aj takúto možnosť a zmerať parametre a taktiež sledovať zmeny pri použití rôznych QoS mechanizmov.

6.1 Testy bez použitia QoS

V tejto časti boli vykonané referenčné testy kde neboli použité žiadne QoS mechanizmy a v sieti bola prípadne nebola generovaná záťaž.

6.1.1 Scenár 1 – Bez QoS a bez záťaže

Výsledky tohto scenáru slúžia čisto na porovnávacie účely. Generovanie záťažových dát bolo vypnuté a prebehla séria testov na oboch typoch liniek (384 Kb/s, 2Mb/s). V týchto testoch samozrejme nebolo možné vykonať posledné dva testy pri sto percentnej záťaži na pozadí. Výsledky sú znázornené v nasledujúcej tabuľke.

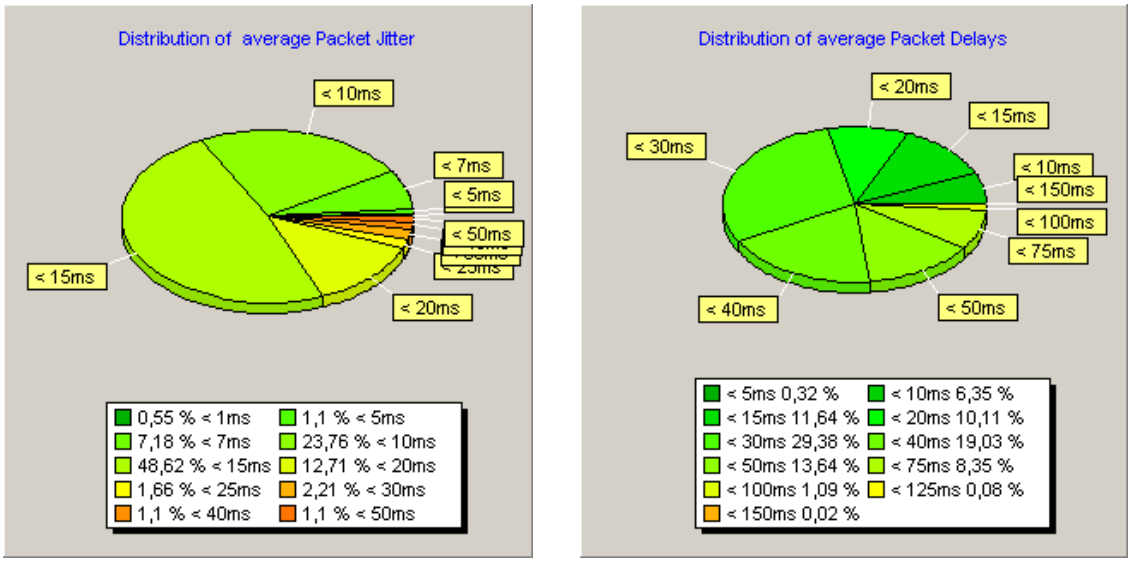
Poznámka: Červené hodnoty v tabuľke signalizujú prekročenie limitnej hodnoty.

Druh testu (rýchlosť linky, hlasové vyťaženie, kodek)	SIP time to connect (ms)	SIP tear down (ms)	Strata rámcov	Strata v %	Pseudo strata	Pseudo strata v %
384kb 60% G.711	21	31	0	0	33	0,2
384kb 90% G.711	31	33	2	0	38	0,1
384kb 60% G.729	21	32	0	0	6	0
384kb 90% G.729	22	188	8230	**10,1**	34	0
2Mb 60% G.711	16	16	6	0	32	0
2Mb 90% G.711	16	16	73	0	8	0
2Mb 60% G.729	0	16	1077	0,3	5	0

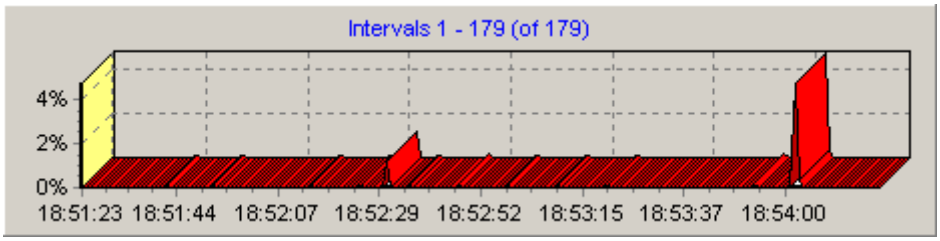
	Preiemer. Oneskorenie (ms)	Max. oneskorenie (ms)	Priemer. Jitter (ms)	Max. Jitter (ms)	R-faktor	MOS
384kb 60% G.711	18	182	7	164	93	4,4
384kb 90% G.711	27	189	9	162	93	4,4
384kb 60% G.729	12	93	5	81	93	4,4
384kb 90% G.729	**221**	459	36	289	**70**	**3,4**
2Mb 60% G.711	13	97	8	84	93	4,4
2Mb 90% G.711	29	136	12	103	93	4,4
2Mb 60% G.729	8	157	4	149	92	4,4

Tab. 6.1.1 -1 Namerané hodnoty pre testovací scenár 1

Ako je z výsledkov vidieť väčšina vykonaných testov prebehla v poriadku, hlasová kvalita sa pohybovala pri maximu. Jediná výnimka bol test s 90 % záťažou s kodekom G.729 kde bola badateľná vysoká strata zapríčinená nedostatkom výpočtového výkonu vysielajúcej a prijímajúcej stanice.
V nasledujúcich grafoch sú zobrazené hodnoty z priebehu testu pri 90 % záťaži hlasom s kodekom G.711.



Obr. 6.1.1 – 1 a 2 Zobrazenie rozloženia hodnôt oneskorenia a jittru v scenári 1



Obr. 6.1.1 – 3 Percentuálne vyjadrenie straty v scenári 1

6.1.2 Scenár 2 – Bez QoS a so záťažou

Tento scenár mal prakticky demonštrovať deštruktívne podmienky pri zmiešaných sieťach (hlas, dáta) v ktorých nie je použitý žiadny QoS mechanizmus. Podľa očakávaní, výsledky tohto scenára mali byť najhoršie a testy s jednotlivými QoS mechanizmami boli porovnávané práve s týmito hodnotami. Ako generátor záťaže bol použitý generátor paketov D-ITG s týmito nastaveniami:

- Veľkost generovaného paketu – 1300 B
- Distribučné rozdelenie – Poissonov tok
- Zdrojový port – 2400
- Cieľový port – 2900
- Trvanie experimentu – 180 sekúnd
- Stredná hodnota toku :
  - 384 Kb/s
    - 30 % zaťaženie – 11 pps (packets per second)
    - 100 % zaťaženie – 37 pps
  - 2Mb/s
    - 30 % zaťaženie – 57 pps (packets per second)
    - 100 % zaťaženie – 192 pps

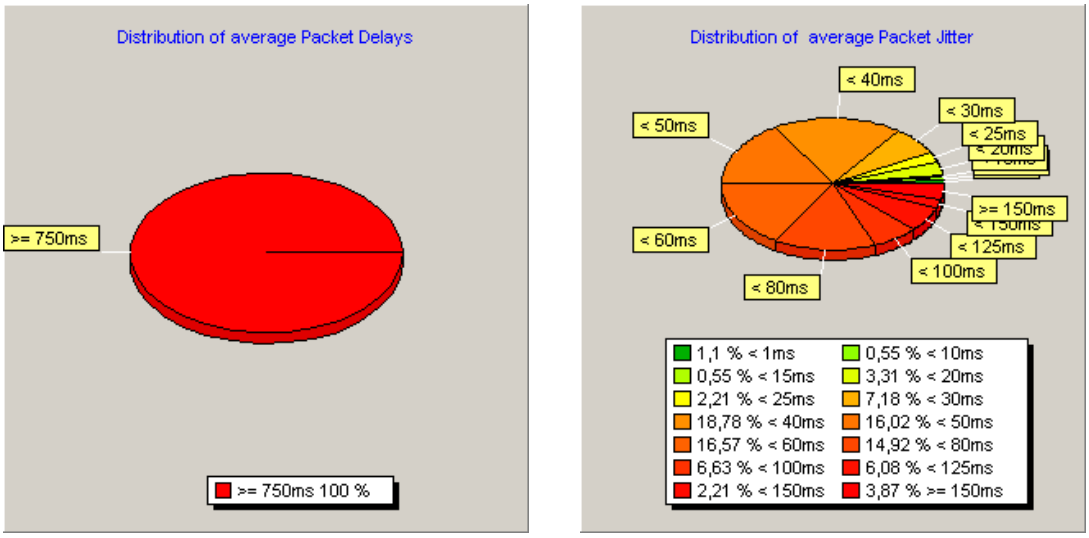
Tieto hodnoty boli stabilné a boli použité vo všetkých testovacích scenároch kde sa na pozadí generovala záťaž.

Výsledky sú znázornené v nasledujúcej tabuľke:

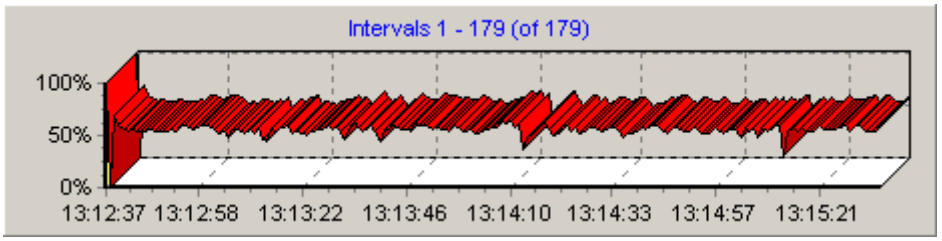
Druh testu (rýchlosť linky, hlasové vyťaženie, kodek, záťaž)	SIP time to connect (ms)	SIP tear down (ms)	Strata rámcov	Strata v %	Pseudo Strata	Pseudo strata v %
384kb 60% G.711 30%	31	220	23	0,1	66	0,4
384kb 90% G.711 30%	32	422	8858	**33,1**	65	0,2
384kb 60% G.729 30%	31	350	3108	**6,9**	52	0,1
384kb 60% G.729 100%	490	1109	24815	**55,6**	123	0,3
384kb 60% G.711 100%	620	2000	9273	**52,2**	236	**1,3**
2Mb 60% G.729 30%	16	47	43813	**10,2**	4	0
2Mb 60% G.711 30%	16	16	176	0,2	3	0
2Mb 90% G.711 30%	16	78	53659	**29,8**	6	0
2Mb 60% G.729 100%	16	370	194204	**58,5**	62	0
2Mb 60% G.711 100%	500	400	57565	**49,4**	39	0
	Preiemer. Oneskorenie (ms)	**Max. oneskorenie (ms)**	**Priemer. Jitter (ms)**	**Max. Jitter (ms)**	**R-faktor**	**MOS**
384kb 60% G.711 30%	**482**	1091	**198**	609	69	**3,3**
384kb 90% G.711 30%	**985**	1435	**147**	948	57	**2,9**
384kb 60% G.729 30%	**415**	871	**115**	456	80	3,8
384kb 60% G.729 100%	**1359**	1883	61	524	57	**2,9**
384kb 60% G.711 100%	**2362**	2730	68	664	47	**2**
2Mb 60% G.729 30%	62	190	12	91	89	4,3
2Mb 60% G.711 30%	17	145	8	128	93	4,4
2Mb 90% G.711 30%	144	324	25	180	87	4,3
2Mb 60% G.729 100%	**237**	435	16	198	57	**2,9**
2Mb 60% G.711 100%	**464**	675	15	213	57	**2,9**

Tab. 6.1.2 – 1 hodnoty pre testovací scenár 2

Ako bolo predpokladané, výsledky parametrov hovorov v tomto scenári ďaleko prevyšovali povolené hodnoty. Hlas mal veľké oneskorenie, taktiež strata bola vysoká. Vysoká strata zapríčinila, že zvuk bol nezrozumiteľný a nepoužiteľný. Tiež signalizácia zaznamenala výrazné oneskorenie oproti nezaťaženej sieti. Nasledujúce grafy znázorňujú podrobnosti hraničného testu pri 100 % záťaži na pozadí a 60 % hlasovom zaťažení kodekom G.729 na 384 Kb/s linke.



Obr. 6.1.2 – 1 a 2 Zobrazenie rozloženia hodnôt oneskorenia a jittru v scenári 2



Obr. 6.1.2 – 3 Percentuálne vyjadrenie straty v scenári 2

6.2 Testy s použitím QoS

V tejto časti boli vykonané testy s jednotlivými QoS mechanizmami ktoré sú k dispozícii v Cisco IOS. Jedná sa o testy s LLQ, LFI, kompresiou IP a RTP a IP RTP Prioritou. Jednotlivé konfigurácie boli nastavené tak, aby bola splnená požiadavka pre maximálnu kvalitu prenášaných hovorov ako bolo navrhnuté v testoch. QoS pre záťažové dáta v týchto testoch neboli brané do úvahy pretože sa zameriavam iba na hlas ale v reálnej aplikácii by boli konfigurácie trochu odlišné, prenosových tried by bolo viacej zahrňujúc aj dátové prenosy teraz považované iba za záťaž.

6.2.1 Scenár 3 – Low Latency Queuing (LLQ)

V tomto scenári som otestoval možnosti konfigurácie LLQ a taktiež jeho praktické nasadenie. Parametre testov a generátorov boli rovnaké ako v predchádzajúcich testoch. Konfigurácia LLQ spočíva avšak v zadefinovaní šírky pásma, ktoré bude pre danú triedu sieťovej prevádzky vyhradené. V mojej konfigurácii som počítal s troma triedami – hlas, signalizácia a default trieda ktorá zabezpečovala zvyšnú sieťovú prevádzku. Šírku pásma som pre jednotlivé testy podelil ako je znázornené v tabuľke.

Dostupná šírka pásma	Rezervované pre hlas	Rezervované pre signalizáciu
384 Kb/s	350 Kb/s	15 Kb/s
2 Mb/s	1823 Kb/s	15 Kb/s

Tab 6.2.1 – 1 Rozdelenie šírky pásma pre jednotlivé triedy

Z tabuľky sa na prvý pohľad zdá byť nelogické, že pri väčšom objeme hovorov na 2 Mb linke je pre signalizáciu rezervovaná tá istá časť pásma – 15 Kb/s avšak je nutné dodať, že táto šírka pásma je rezervovaná kontinuálne a signalizácia tak ako ju testujem ja taktiež prebieha kontinuálne. V prípade väčšieho náporu SIP paketov by bola táto komunikácia spomalená v rádovo milisekundách v dôsledku malej veľkosti paketov. Taktiež z hodnôt šírky pásma rezervovaných pre hlas (RTP pakety) je badateľné, že pri niektorých testoch táto šírka pásma nemusí byť dostatočná. To sa samozrejme odzrkadlí na hodnotách straty paketov keďže prioritné fronty budú preplnené a prebytočné pakety sa budú zahadzovať. Tento problém je avšak riešiteľný vhodným dimenzovaním linky, nastavením tried a počtom IP telefónov ktoré sú v sieti.

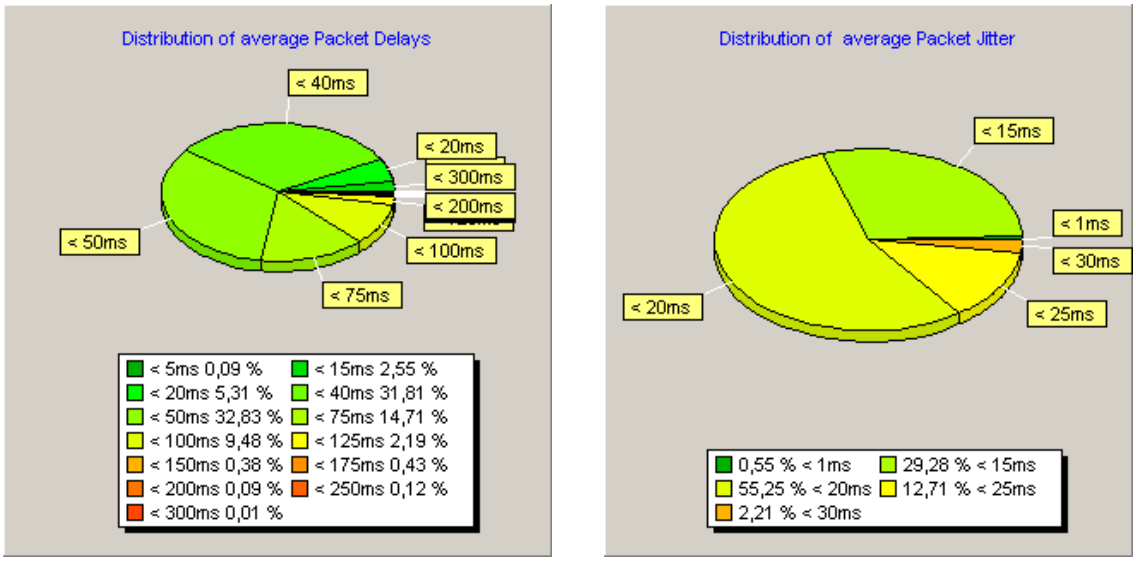
Výsledky testov daného scenára reprezentuje nasledovná tabuľka:

Druh testu (rýchlosť linky, hlasové vyťaženie, kodek, záťaž)	SIP time to connect (ms)	SIP tear down (ms)	Strata rámcov	Strata v %	Pseudo Strata	Pseudo strata v %
384Kb 60% G.711 100%	55	141	5	0	478	**2,7**
384Kb 60% G.729 100%	60	160	1	0	198	0,4
384Kb 60% G.729 30%	50	110	1	0	206	0,5
384Kb 90% G.711 30%	200	800	1135	**3,2**	42	0,2
384Kb 60% G.711 30%	47	190	0	0	163	0,9
2Mb 60% G.711 30%	16	15	83	0,1	185	0,2
2Mb 90% G.711 30%	16	15	14905	**7,4**	240	0,1
2Mb 60% G.729 30%	15	16	10385	2,1	234	0,1
2Mb 60% G.729 100%	16	22	26003	**6,8**	202	0,1
2Mb 60% G.711 100%	15	21	21122	**4,1**	167	0,1

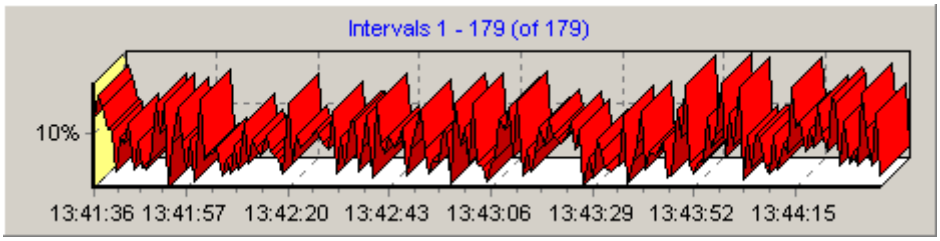
	Preiemer. Oneskorenie (ms)	Max. oneskorenie (ms)	Priemer. Jitter (ms)	Max. Jitter (ms)	R-faktor	MOS
384Kb 60% G.711 100%	81	233	19	152	93	4,4
384Kb 60% G.729 100%	74	185	18	111	93	4,4
384Kb 60% G.729 30%	73	258	18	185	93	4,4
384Kb 90% G.711 30%	77	207	18	130	84	4,1
384Kb 60% G.711 30%	69	239	16	170	93	4,4
2Mb 60% G.711 30%	32	249	13	217	93	4,4
2Mb 90% G.711 30%	108	312	18	188	87	4,2
2Mb 60% G.729 30%	47	265	16	218	93	4,4
2Mb 60% G.729 100%	43	234	16	144	92	4,4
2Mb 60% G.711 100%	34	218	14	184	93	4,4

Tab. 6.2.1 – 2 hodnoty pre testovací scenár 3

Ako je z tabuľky vidieť hodnoty kvality si výrazne polepšili oproti testu bez QoS. Samozrejme aj tu nastala v niektorých prípadoch vyššia strata paketov, ale to bolo spôsobené nastavaním triedy hlasu na nižšiu hodnotu bolo nutné pre úplné pokrytie. Avšak ani strata 7% ktorá nastala v jednom z testov nijako výrazne neovplyvnila celkovú kvalitu prenášaných hovorov keďže hodnoty oneskorenia, jittru a oneskorenia signalizácie boli v norme. Nasledujúce grafy znázorňujú podrobnosti hraničného testu pri 30 % záťaži na pozadí a 60 % hlasovom zaťažení kodekom G.729 na 2 Mb/s linke.



Obr. 6.2.1 – 1 a 2 Zobrazenie rozloženia hodnôt oneskorenia a jittru v scenári 3



Obr. 6.2.1 – 3 Percentuálne vyjadrenie straty v scenári 3

6.2.2 Scenár 4 – Low Latency Queuing (LLQ) s LFI a cRTP

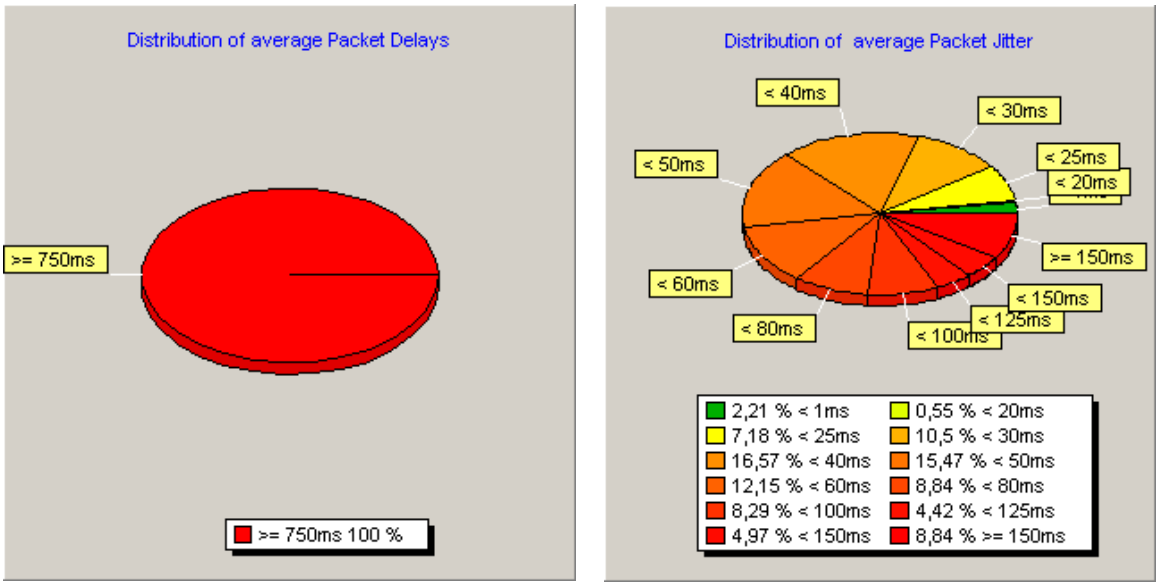
Tento scenár bol zameraný na otestovanie všetkých dostupných metód na zabezpečenie lepšej kvality hlasu. Pri tomto scenáre sa avšak prejavil istý paradox kedy namerané hodnoty boli o niečo horšie ako v predchádzajúcom scenári kde bolo použité iba LLQ radenie do frontov. Toto zhoršenie ma na svedomí prílišné zaťaženie procesorov smerovačov keďže musel dodatočne komprimovať veľké množstvo dát (všetky IP pakety a dodatočne ešte RTP hlavičky) a zároveň deliť a prekladať veľké dátové pakety generované na pozadí. Toto delenie bolo navyše nastavené tak, aby oneskorenie paketov nebolo väčšie ako 20 milisekúnd. Všetky tieto faktory pravdepodobne spôsobili mierne zhoršenie parametrov hovorov oproti predchádzajúcemu scenáru. Všeobecné pravidlo pre kompresiu vraví, že by sa nemala používať na linkách s priepustnosťou väčšou ako 1Mb/s, ja som toto pravidlo porušil a výsledky nasvedčujú, že výpočty z dielne cisco boli správne. Pre tento fakt v tabuľke výsledkov ani neuvádzam hodnoty pre merania hraničnej záťaže pri viac ako 3,2 Mb/s na 2Mb/s linke.

Výsledky testov daného scenára reprezentuje nasledovná tabuľka:

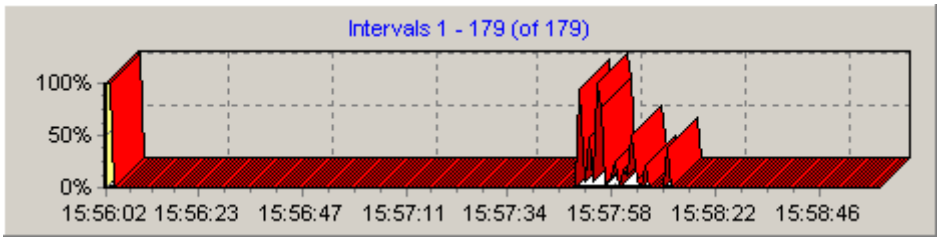
Druh testu (rýchlosť linky, hlasové vyťaženie, kodek, záťaž)	SIP time to connect (ms)	SIP tear down (ms)	Strata rámcov	Strata v %	Pseudo Strata	Pseudo strata v %
2Mb 60% G.711 30%	7	16	84	0,1	174	0,1
2Mb 90% G.711 30%	6	150	42012	**24**	382	0,2
2Mb 60% G.729 30%	17	30	20559	**6,2**	229	0,1
384Kb 60% G.711 30%	30	41	0	0	89	0,5
384Kb 90% G.711 30%	30	710	2	0	132	0,5
384Kb 60% G.729 30%	32	49	0	0	173	0,4
384Kb 60% G.729 100%	46	750	2110	**4,7**	472	1,1
384Kb 60% G.711 100%	47	1290	640	**3,6**	209	1,2
	Preiemer. Oneskorenie (ms)	**Max. oneskorenie (ms)**	**Priemer. Jitter (ms)**	**Max. Jitter (ms)**	**R-faktor**	**MOS**
2Mb 60% G.711 30%	28	249	16	174	93	4,4
2Mb 90% G.711 30%	**168**	421	19	253	78	3,9
2Mb 60% G.729 30%	40	265	17	178	92	4,4
384Kb 60% G.711 30%	50	186	17	136	93	4,4
384Kb 90% G.711 30%	**263**	643	56	623	89	4,3
384Kb 60% G.729 30%	44	218	17	174	93	4,4
384Kb 60% G.729 100%	**512**	1101	**210**	234	78	3,9
384Kb 60% G.711 100%	**453**	1209	56	345	88	4,2

Tab. 6.2.2 – 1 hodnoty pre testovací scenár 4

Hodnoty z tabuľky ukazujú zvýšenú mieru oneskorenia a straty paketov čo môže byť spôsobené práve nedostatočným výkonom smerovačov pri fragmentácii spoločne s kompresiou. Nasledujúce grafy znázorňujú podrobnosti hraničného testu pri 100 % záťaži na pozadí a 60 % hlasovom zaťažení kodekom G.711 na 384 Kb/s linke.



Obr. 6.2.2 – 1 a 2 Zobrazenie rozloženia hodnôt oneskorenia a jittru v scenári 4



Obr. 6.2.2 – 3 Percentuálne vyjadrenie straty v scenári 4

Poznámka: Z grafu, ktorý zobrazuje stratu paketov v čase je vidieť istý časový interval kedy strata paketov dosahuje až 100 %. V priebehu testovania sa mi avšak nepodarilo zistiť kde v sieti presne nastal problém ale predpokladám, že mohol nastať problém na niektorej zo staníc, ktoré mirrorovali RTP pakety. Tieto stanice nie sú moc výkonné a prípadná aktualizácia operačného systému môže spôsobiť výrazné spomalenie celého systému a tým aj skreslenie výsledkov.

6.2.3 Scenár 5 – IP RTP Priority

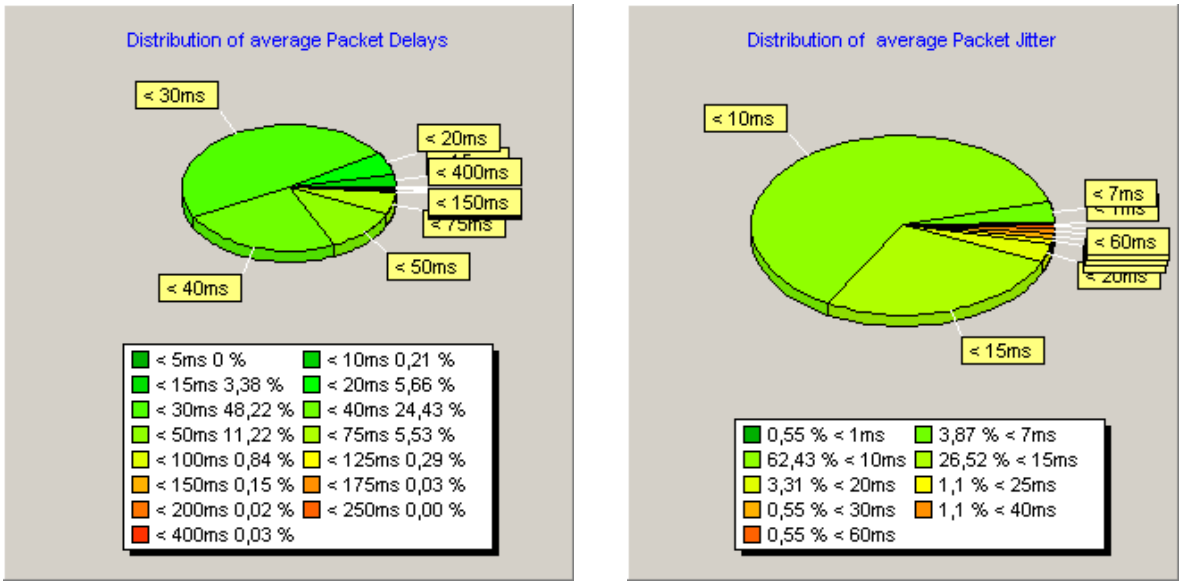
IP RTP priority je jednoduchý systém, kedy je istý typ paketov uprednostňovaný na výstupnej fronte. Keďže konfigurácia je veľmi jednoduchá a očakávanie úspešnosti nízke, výsledok o to viacej prekvapil. V podstate konfigurácia IP RTP Priority prebieha tak, že sa zvolí istý rozsah portov a šírka pásma ktorá bude týmto dátam vždy k dispozícii. Pre 384 Kb/s linku bol nakonfigurovaný prioritný front na 350 Kb/s a pri 2 Mb/s linke bola priorita nastavaná na 1544 Kb/s, čo bolo maximu čo IOS pri danej konfigurácii povoľoval. Toto samozrejme v istým prípadoch nemohlo stačiť, čo sa prejavilo najmä pri 90 % zaťažení siete hlasom. V ostatných prípadoch avšak dané nastavenia vyhovovali a parametre hovorov boli v norme.

Výsledky testov daného scenára reprezentuje nasledovná tabuľka:

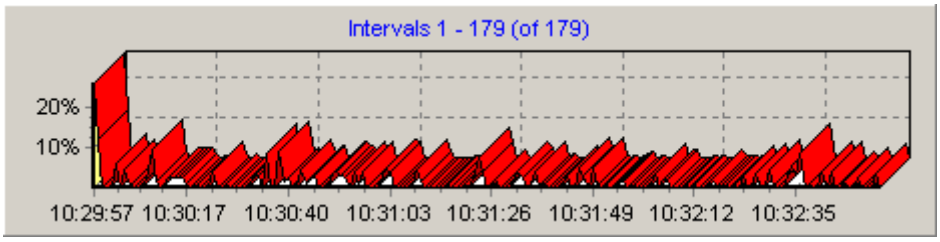
Druh testu (rýchlosť linky, hlasové vyťaženie, kodek, záťaž)	SIP time to connect (ms)	SIP tear down (ms)	Strata rámcov	Strata v %	Pseudo Strata	Pseudo strata v %
384Kb 60% G.711 30%	62	47	2	0	183	1
384Kb 90% G.711 30%	90	31	29	0,1	2	0
384Kb 60% G.729 30%	60	100	0	0	146	0,3
384Kb 60% G.729 100%	76	180	0	0	149	0,3
384Kb 60% G.711 100%	62	146	0	0	266	1,5
2Mb 60% G.711 30%	16	31	9	0	27	0
2Mb 90% G.711 30%	16	100	35656	18	221	0,1
2Mb 60% G.729 30%	16	30	3947	1,2	26	0
2Mb 60% G.729 100%	16	78	4863	1,5	37	0
2Mb 60% G.711 100%	16	74	5	0	25	0
	Preiemer. Oneskorenie (ms)	**Max. oneskorenie (ms)**	**Priemer. Jitter (ms)**	**Max. Jitter (ms)**	**R-faktor**	**MOS**
384Kb 60% G.711 30%	64	195	20	131	93	4,4
384Kb 90% G.711 30%	26	200	8	174	93	4,4
384Kb 60% G.729 30%	69	228	18	159	93	4,4
384Kb 60% G.729 100%	72	197	18	125	93	4,4
384Kb 60% G.711 100%	76	228	17	152	93	4,4
2Mb 60% G.711 30%	17	210	7	193	93	4,4
2Mb 90% G.711 30%	40	325	13	285	78	3,9
2Mb 60% G.729 30%	29	170	8	141	93	4,4
2Mb 60% G.729 100%	31	347	10	316	93	4,4
2Mb 60% G.711 100%	25	211	4	186	93	4,4

Tab. 6.2.3 – 1 hodnoty pre testovací scenár 5

Z výsledkov vidieť že všetky testy prebehli v poriadku až na test s 90% záťažou pri 2 Mb/s linke. Hodnoty signalizácie síce nebolo možné ovplyvniť, a z výsledkov je zrejmé kolísanie hodnôt oneskorení SIP paketov, ale tieto sú nižšie ako pri pokusoch bez QoS mechanizmov. Veľká nevýhoda tejto metódy je jednostranné zameranie a možnosť klasifikácie paketov len na základe rozsahu portov. Avšak treba dodať, že táto funkcia by v 90 % hlasových sietí vykonala dobrú prácu. V komplexnejších sieťach ale môžeme naraziť na problém, že máme len jeden front pre hlas a signalizáciu nechávame bokom. Nasledujúce grafy znázorňujú podrobnosti testu pri 100 % záťaži na pozadí a 60 % hlasovom zaťažení kodekom G.729 na 2 Mb/s linke.



Obr. 6.2.3 – 1 a 2 Zobrazenie rozloženia hodnôt oneskorenia a jittru v scenári 5



Obr. 6.2.3 – 3 Percentuálne vyjadrenie straty v scenári 5

7. Hodnotenie a záver

V rámci tejto diplomovej práce som popisoval, analyzoval a testoval dopad použitia rôznych mechanizmov, ktoré zabezpečujú kvalitu hlasu v sieti, po ktorej sú smerované dáta rôznych typov. Ako bolo pri praktickom testovaní dokázané, je veľké riziko nasadiť do podnikových prípadne iných typov sietí riešenia IP telefónie bez dodatočnej podpory infraštruktúry. QoS mechanizmy by mali byť integrálnou súčasťou sietí v ktorých sa prenáša digitálne spracovaný hlas, video prípadne iný typ real-time prenosu. Je tiež ale výhodné tieto mechanizmy použiť vo všetkých sieťach kde sa očakáva zabezpečenie služieb na inej úrovni ako best effort.

Čo sa týka jednotlivých mechanizmov ktoré som testoval, najperspektívnejšia sa zdá byť cesta použitia class–based QoS mechanizmov akým je LLQ. Podľa potrieb sa môžu parametre tried a priority nastaviť tak, aby presne vyhovovali danému počtu užívateľov, dátovému prenosu a počtu telefónov v sieti. Aj napriek zložitejšej konfigurácii odporúčam preto LLQ pred IP RTP Priority pretože IP RTP Priority nedokáže zabezpečiť dokonalé oddelenie dát ktoré smerujú do prioritného frontu, nie je možné vytvoriť ďalší prioritný front pre iný typ senzitívnych dát čo môže byť veľkou nevýhodou hlavne pre posielanie videa sieťou. V prípade, že by sme do prioritného frontu zároveň púšťali aj hlas aj video, ktoré je menej senzitívne na oneskorenie a jitter, mohla by nastať situácia kedy by oba typy real-time prenosov mali výrazne zhoršené parametre kvality. Použitie LFI a kompresie je sporadickým bodom a závisí hlavne na priepustnosti linky, strednej veľkosti dátových paketov, ktoré cez sieť prenášame a taktiež od použitých smerovačov v sieti. V niektorých prípadoch sa oplatí LFI a kompresiu nepoužiť, čím sa zabezpečí celková vyššia priepustnosť v uzloch siete. Taktiež použitím call admission kontroly, ktorá obmedzí ďalšie hovory v sieti keď sa vyčerpajú zdroje, v podstate chránime hlas pred hlasom.
V rámci testovania som predstavil viacero typov softvérov, ktoré monitorovali, pasívne alebo aktívne testovali kvalitu generovaného hlasu v sieti. Výrazným spôsobom mi pri tom pomohol softvér NetQuality, ktorý síce nie celkom zodpovedal reálnemu testovaniu hlasu v sieti, zato ale poskytol detailné výsledky a grafické znázornenie. V oblasti signalizácie by som vyzdvihol softvér WinSIP ktorý je profesionálny nástroj používaný na testovanie v rôznych veľkých spoločnostiach akými sú napr. AT&T, Siemens alebo Ericsson.

V každej dnešnej projektovanej podnikovej, školskej sieti alebo v štátnej správe je nutné počítať aj s nasadením hlasových služieb. Keďže technológie na prenos hlasu cez IP sú v porovnaní s hlasovými službami založenými na prepínaných okruhoch lacnejšie a nevyžadujú dodatočnú kabeláž, bude do budúcnosti rásť dopyt po takomto type služieb. Ako príklad môžu slúžiť dnešné call centrá na Slovensku v ktorých je podľa analýzy 80% hlasových služieb založených na VoIP. Zabezpečenie kvality tak multimédií ako samotných dátových prenosov je veľká téma a treba s tým počítať už pri samom dizajne navrhovanej siete. Je všeobecný predpoklad, že v budúcnosti budú tieto technológie nasadzované aj v oblasti dnešných bezdrôtových sietí pre zabezpečenie video hovorov prípadne pre samotné hovory, ktoré by úplne migrovali na paketovú technológiu. Táto diplomová práca teda poskytla komplexný pohľad na problematiku, druhy riešenia a praktické nasadenie do existujúcej siete. Boli rozobraté techniky merania a dostupné nástroje pomocou ktorých môžeme dostatočne kvalitne otestovať existujúce aj navrhované siete. Logickým výsledkom by teda mala byť zvýšená produktivita a komfort práce.

8. Zoznam skratiek

A

ALG	Application Layer Gateway
ATM	Asynchronous Transfer Mode
ACL	Access List

B

B2BUA	Back-To-Back User Agent

C

CRLF	Carriage-Return Line-Feed Sequence
CAC	Call Admission Control
COPS	Common Open Policy Service
CoS	Class of Service
CAR	Commited Access Rate
CBWFQ	Class Based Weighted Fair Queuing
CEF	Cisco Expess Forwarding
CRTP	Compressed Real Time Protocol
CIR	Commited Information Rate

D

DSL	Digital Subscriber Line
DSP	Digital Signal Processor
DiffServ	Differentiated Services
DSCP	Differentiated Services Code Point

F

FIFO	First In First Out
FRF.12	Frame Relay Forum 12

H

H.323	Sada protokolov používaných na signalizáciu
HTTP	Hypertext Transfer Protocol

I

IAX	Inter Asterisk Exchange
IETF	Internet Engineering Task Force
ITU	International Telecomunication Union
ITU-T	ITU – Telecomunication Standardization Sector
IP	Internet Protocol
IntServ	Integrated Services
IOS	Internetwork Operating System

L

LAN	Local Area Network

LDAP	Lightweight Directory Access Protocol
LMOS	Listening Mean Opinion Score
LQ	Listening Quality
LLQ	Low Latency Queuing

M

MCU	Multipoint Control Unit
MBONE	Multicast Backbone
MGCP	Media Gateway Control Protocol
MEGACO	Media Gateway Control
MOS	Mean Opinion Score
MLP	Multilink PPP
MTU	Maximum Transmission Unit

P

PSTN	Public Switched Telephone Network
PESQ	Perceptual Evaluation of Speech Quality
PBX	Private Branch Exchange
PHB	Per Hop Behavior
PQ	Priority Queue

Q

QoS	Quality Of Service

R

RCNA	Regional Cisco Networking Academy
RTP	Real Time Protocol
RTCP	Real Time Control Protocol
RSVP	Resource Reservation Protocol
RFC	Request For Comments
RTCP-XR	RTP Control Protocol Extended Reports

S

SIP	Session Initiation Protocol
SS7	Signaling System 7
SCCP	Skinny Client Control Protocol
SOAP	Simple Object Access Protocol
SDP	Session Description Protocol
SMTP	Simple Mail Transfer Protocol
SLA	Service Level Agreement

T

TCP	Transmission Control Protocol
TDM	Time Division Multiplexing
TTL	Time To Live
ToS	Type of Service
TCA	Traffic Conditionig Agreement

U

UDDI	Universal Description Discovery and Integration
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
UAC	User Agent Client
UAS	User Agent Server
UTF-8	Unicode Transformation Format 8

V

VoIP	Voice Over IP
VXML	Voice Extensible Markup Language
VAD	Voice Activity Detection
VQES	Voice Quality Evaluation

W

WAN	Wide Are
WSDL	Web Service Definition Language
WFQ	Weighted Fair Queuing
WRED	Weighted Random Early Detection

X

XML	Extensible Markup Language

Y

YESSIR	Yet another Sender Session Internet Reservations

9. Prílohy

9.1 Konfigurácia klasifikácie smerovačov R1 a R5

(Tieto konfigurácie sú zhodné, uvádzam konfiguráciu pre R1. Ostatné konfigurácie sa nachádzajú na doprovodnom CD)

```
service timestamps debug datetime
msec
service timestamps log datetime msec
no service password-encryption
!
hostname R1
!
boot-start-marker
boot-end-marker
!
mmi polling-interval 60
no mmi auto-configure
no mmi pvc
mmi snmp-timeout 180
no aaa new-model
ip subnet-zero
ip cef
!
ip dhcp pool pool110
   network 192.168.110.0
255.255.255.0
   domain-name alpha
   default-router 192.168.110.1
!
ip ips po max-events 100
no ftp-server write-enable
!
class-map match-all signaling-marking
 match access-group 103
class-map match-all voice-marking
 match access-group 102
!
policy-map setdscp
 class voice-marking
  set dscp ef
 class signaling-marking
  set dscp cs3
!
interface FastEthernet0/0
 description "eth subnet 110.0"
 ip address 192.168.110.1
255.255.255.0
 service-policy input setdscp
 duplex auto
 speed auto
 no shutdown
!
interface FastEthernet0/1
 description eth na r3
 ip address 192.168.3.1 255.255.255.0
 duplex auto
 speed auto
 no shutdown
!
router rip
 version 2
 network 192.168.3.0
 network 192.168.110.0
!
ip classless
!
ip http server
no ip http secure-server
!
access-list 102 permit udp any any
range 8700 32767
access-list 103 permit udp any any
range 5060 5063
!
control-plane
!
line con 0
line aux 0
line vty 0 4
!
End
```

9.2 Konfigurácia prepínača napojeného na VQManager

```
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Switch
!
!
no aaa new-model
ip subnet-zero
!

!
no file verify auto
spanning-tree mode pvst
spanning-tree extend system-id
!
vlan internal allocation policy
ascending
!
!
interface FastEthernet0/1
!
interface FastEthernet0/2
!
interface FastEthernet0/3
!
interface FastEthernet0/4
!
interface FastEthernet0/5
!
interface FastEthernet0/6
!
interface FastEthernet0/7
!
interface FastEthernet0/8
!
interface FastEthernet0/9
!
interface FastEthernet0/10
!
interface FastEthernet0/11
!
interface FastEthernet0/12
!
interface FastEthernet0/13
!
interface FastEthernet0/14
!
interface FastEthernet0/15
!
interface FastEthernet0/16
!
interface FastEthernet0/17
!
interface FastEthernet0/18
!
interface FastEthernet0/19
!
interface FastEthernet0/20
!
interface FastEthernet0/21
!
interface FastEthernet0/22
!
interface FastEthernet0/23
!
interface FastEthernet0/24
!
interface GigabitEthernet0/1
!
interface GigabitEthernet0/2
!
interface Vlan1
 ip address dhcp
!
ip classless
ip http server
!
!
!
control-plane
!
!
line con 0
line vty 0 4
 no login
line vty 5 15
no login
!
!
monitor session 1 source interface
Fa0/1 , Fa0/5
monitor session 1 destination
interface Fa0/24
!
end
```

9.3 Konfigurácia smerovača R3

9.3.1 – R3 s LLQ

```
service timestamps debug datetime
msec
service timestamps log datetime msec
no service password-encryption
!
hostname R3
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
!
resource policy
!
ip subnet-zero
!
!
ip cef
ip cef accounting per-prefix non-
recursive
ip cef traffic-statistics update-rate
1
no ip dhcp use vrf connected
!
!
no ip ips deny-action ips-interface
!
!
!
!
!
!
class-map match-all voice-signaling
 match ip dscp cs3
class-map match-all voice-traffic
 match ip dscp ef
!
!
policy-map voice-policy
 class voice-traffic
  priority 350
 class voice-signaling
  bandwidth 15
 class class-default
  fair-queue
!
!
!
interface FastEthernet0/0
 description monitorovaci interface
 ip address 192.168.100.1
255.255.255.0
 ip route-cache flow
 duplex auto
 speed auto
 no shutdown
!
interface FastEthernet0/1
 description "link na R1"
 ip address 192.168.3.2 255.255.255.0
 duplex auto
 speed auto
 no shutdown
!
interface Serial0/0/0
 description "link na r4"
 ip address 192.168.12.1
255.255.255.0
 ip route-cache flow
 service-policy output voice-policy
 no shutdown
!
interface Serial0/0/1
 description "s1 link na R2"
 ip address 192.168.6.2 255.255.255.0
 clockrate 2000000
 no shutdown
!
!
router rip
 version 2
 network 192.168.3.0
 network 192.168.6.0
 network 192.168.12.0
!
ip classless
!
ip flow-export source FastEthernet0/0
ip flow-export destination
192.168.100.2 2055
!
ip http server
no ip http secure-server
!
snmp-server community public RW
snmp-server community testsip RW
snmp-server enable traps snmp
authentication linkdown linkup
coldstart warmstart
snmp-server enable traps vrrp
snmp-server enable traps tty
snmp-server enable traps eigrp
snmp-server enable traps xgcp
snmp-server enable traps flash
insertion removal
snmp-server enable traps ds3
snmp-server enable traps envmon
snmp-server enable traps icsudsu
snmp-server enable traps isdn call-
information
snmp-server enable traps isdn layer2
snmp-server enable traps isdn chan-
not-avail
snmp-server enable traps isdn ietf
snmp-server enable traps ds0-busyout
snmp-server enable traps ds1-loopback
snmp-server enable traps atm subif
snmp-server enable traps bgp
snmp-server enable traps bulkstat
collection transfer
snmp-server enable traps cnpd
snmp-server enable traps config-copy
snmp-server enable traps config
snmp-server enable traps dial
```

```
snmp-server enable traps dsp card-
status
snmp-server enable traps entity
snmp-server enable traps event-
manager
snmp-server enable traps frame-relay
snmp-server enable traps frame-relay
subif
snmp-server enable traps hsrp
snmp-server enable traps ipmobile
snmp-server enable traps ipmulticast
snmp-server enable traps mpls ldp
snmp-server enable traps mpls
traffic-eng
snmp-server enable traps mpls vpn
snmp-server enable traps msdp
snmp-server enable traps mvpn
snmp-server enable traps ospf state-
change
snmp-server enable traps ospf errors
snmp-server enable traps ospf
retransmit
snmp-server enable traps ospf lsa
snmp-server enable traps ospf cisco-
specific state-change nssa-trans-
change
snmp-server enable traps ospf cisco-
specific state-change shamlink
interface-old
snmp-server enable traps ospf cisco-
specific state-change shamlink
neighbor
snmp-server enable traps ospf cisco-
specific errors
snmp-server enable traps ospf cisco-
specific retransmit
snmp-server enable traps ospf cisco-
specific lsa
snmp-server enable traps pim
neighbor-change rp-mapping-change
invalid-pim-message
snmp-server enable traps pppoe
snmp-server enable traps cpu
threshold
snmp-server enable traps rsvp
snmp-server enable traps rtr
snmp-server enable traps syslog
snmp-server enable traps l2tun
session
snmp-server enable traps vsimaster
snmp-server enable traps vtp
snmp-server enable traps isakmp
policy add
snmp-server enable traps isakmp
policy delete
snmp-server enable traps isakmp
tunnel start
snmp-server enable traps isakmp
tunnel stop
snmp-server enable traps ipsec
cryptomap add
snmp-server enable traps ipsec
cryptomap delete
snmp-server enable traps ipsec
cryptomap attach
snmp-server enable traps ipsec
cryptomap detach
snmp-server enable traps ipsec tunnel
start
snmp-server enable traps ipsec tunnel
stop
snmp-server enable traps ipsec too-
many-sas
snmp-server enable traps voice poor-
qov
snmp-server enable traps voice
fallback
snmp-server enable traps dnis
snmp-server host 192.168.100.2 public
!
!
!
!
control-plane
!
!
!
!
!
!
!
!
!
line con 0
line aux 0
line vty 0 4
 login
!
scheduler allocate 20000 1000
!
end
```

9.3.2 – R3 s LFI, LLQ a cRTP

```
service timestamps debug datetime
msec
service timestamps log datetime msec
no service password-encryption
!
hostname R3
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
!
resource policy
!
ip subnet-zero
!
!
ip cef
ip cef accounting per-prefix non-
recursive
ip cef traffic-statistics update-rate
1
no ip dhcp use vrf connected
!
!
no ip ips deny-action ips-interface
!
!
!
!
!
!
class-map match-all voice-signaling
 match ip dscp cs3
class-map match-all voice-traffic
 match ip dscp ef
!
!
policy-map voice-policy
 class voice-traffic
  priority 1823
 class voice-signaling
  bandwidth 15
 class class-default
  fair-queue
!
interface Multilink1
 ip address 192.168.12.1
255.255.255.0
 ip tcp header-compression iphc-
format
 service-policy output voice-policy
 no cdp enable
 no ip direct broadcast
 ppp multilink
 ppp multilink fragment-delay 20
 ppp multilink interleave
 multilink-group 1
  ip rtp header-compression iphc-
format
 max-reserved-bandwidth 100
!
!
interface FastEthernet0/0
 description monitorovaci interface
 ip address 192.168.100.1
255.255.255.0
 ip route-cache flow
 duplex auto
 speed auto
 no shutdown
!
interface FastEthernet0/1
 description "link na R1"
 ip address 192.168.3.2 255.255.255.0
 duplex auto
 speed auto
 no shutdown
!
interface Serial0/0/0
 bandwidth 2000
 description "link na r4 s LFI"
 no ip address
 encapsulation ppp
 ppp multilink
 multilink group 1
 no fair queue
 no shutdown
!
interface Serial0/0/1
 description "s1 link na R2"
 ip address 192.168.6.2 255.255.255.0
 clockrate 2000000
 no shutdown
!
!
router rip
 version 2
 network 192.168.3.0
 network 192.168.6.0
 network 192.168.12.0
!
ip classless
!
ip flow-export source FastEthernet0/0
ip flow-export destination
192.168.100.2 2055
!
ip http server
no ip http secure-server
!
snmp-server community public RW
snmp-server community testsip RW
snmp-server enable traps snmp
authentication linkdown linkup
coldstart warmstart
snmp-server enable traps vrrp
snmp-server enable traps tty
snmp-server enable traps eigrp
snmp-server enable traps xgcp
snmp-server enable traps flash
insertion removal
snmp-server enable traps ds3
snmp-server enable traps envmon
snmp-server enable traps icsudsu
snmp-server enable traps isdn call-
information
snmp-server enable traps isdn layer2
snmp-server enable traps isdn chan-
not-avail
snmp-server enable traps isdn ietf
snmp-server enable traps ds0-busyout
```

```
snmp-server enable traps ds1-loopback
snmp-server enable traps atm subif
snmp-server enable traps bgp
snmp-server enable traps bulkstat
collection transfer
snmp-server enable traps cnpd
snmp-server enable traps config-copy
snmp-server enable traps config
snmp-server enable traps dial
snmp-server enable traps dsp card-
status
snmp-server enable traps entity
snmp-server enable traps event-
manager
snmp-server enable traps frame-relay
snmp-server enable traps frame-relay
subif
snmp-server enable traps hsrp
snmp-server enable traps ipmobile
snmp-server enable traps ipmulticast
snmp-server enable traps mpls ldp
snmp-server enable traps mpls
traffic-eng
snmp-server enable traps mpls vpn
snmp-server enable traps msdp
snmp-server enable traps mvpn
snmp-server enable traps ospf state-
change
snmp-server enable traps ospf errors
snmp-server enable traps ospf
retransmit
snmp-server enable traps ospf lsa
snmp-server enable traps ospf cisco-
specific state-change nssa-trans-
change
snmp-server enable traps ospf cisco-
specific state-change shamlink
interface-old
snmp-server enable traps ospf cisco-
specific state-change shamlink
neighbor
snmp-server enable traps ospf cisco-
specific errors
snmp-server enable traps ospf cisco-
specific retransmit
snmp-server enable traps ospf cisco-
specific lsa
snmp-server enable traps pim
neighbor-change rp-mapping-change
invalid-pim-message
snmp-server enable traps pppoe
snmp-server enable traps cpu
threshold
snmp-server enable traps rsvp
snmp-server enable traps rtr
snmp-server enable traps syslog
snmp-server enable traps l2tun
session
snmp-server enable traps vsimaster
snmp-server enable traps vtp
snmp-server enable traps isakmp
policy add
snmp-server enable traps isakmp
policy delete
snmp-server enable traps isakmp
tunnel start
snmp-server enable traps isakmp
tunnel stop
snmp-server enable traps ipsec
cryptomap add
snmp-server enable traps ipsec
cryptomap delete
snmp-server enable traps ipsec
cryptomap attach
snmp-server enable traps ipsec
cryptomap detach
snmp-server enable traps ipsec tunnel
start
snmp-server enable traps ipsec tunnel
stop
snmp-server enable traps ipsec too-
many-sas
snmp-server enable traps voice poor-
qov
snmp-server enable traps voice
fallback
snmp-server enable traps dnis
snmp-server host 192.168.100.2 public
!
!
!
!
control-plane
!
!
!
!
!
!
!
!
line con 0
line aux 0
line vty 0 4
 login
!
scheduler allocate 20000 1000
!
end
```

9.3.3 – R3 s IP RTP Priority

```
service timestamps debug datetime
msec
service timestamps log datetime msec
no service password-encryption
!
hostname R3
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
!
resource policy
!
ip subnet-zero
!
!
ip cef
ip cef accounting per-prefix non-
recursive
ip cef traffic-statistics update-rate
1
no ip dhcp use vrf connected
!
!
no ip ips deny-action ips-interface
!
!
!
!
!
!
interface FastEthernet0/0
 description monitorovaci interface
 ip address 192.168.100.1
255.255.255.0
 ip route-cache flow
 duplex auto
 speed auto
 no shutdown
!
interface FastEthernet0/1
 description "link na R1"
 ip address 192.168.3.2 255.255.255.0
 duplex auto
 speed auto
 no shutdown
!
interface Serial0/0/0
 ip rtp priority 8000 16000 350
 description "link na r4"
 ip address 192.168.12.1
255.255.255.0
 ip route-cache flow
 fair-queue
 no shutdown
!
interface Serial0/0/1
 description "s1 link na R2"
 ip address 192.168.6.2 255.255.255.0
 clockrate 2000000
 no shutdown
!
!
router rip
 version 2
 network 192.168.3.0
 network 192.168.6.0
 network 192.168.12.0
!
ip classless
!
ip flow-export source FastEthernet0/0
ip flow-export destination
192.168.100.2 2055
!
ip http server
no ip http secure-server
!
snmp-server community public RW
snmp-server community testsip RW
snmp-server enable traps snmp
authentication linkdown linkup
coldstart warmstart
snmp-server enable traps vrrp
snmp-server enable traps tty
snmp-server enable traps eigrp
snmp-server enable traps xgcp
snmp-server enable traps flash
insertion removal
snmp-server enable traps ds3
snmp-server enable traps envmon
snmp-server enable traps icsudsu
snmp-server enable traps isdn call-
information
snmp-server enable traps isdn layer2
snmp-server enable traps isdn chan-
not-avail
snmp-server enable traps isdn ietf
snmp-server enable traps ds0-busyout
snmp-server enable traps ds1-loopback
snmp-server enable traps atm subif
snmp-server enable traps bgp
snmp-server enable traps bulkstat
collection transfer
snmp-server enable traps cnpd
snmp-server enable traps config-copy
snmp-server enable traps config
snmp-server enable traps dial
snmp-server enable traps dsp card-
status
snmp-server enable traps entity
snmp-server enable traps event-
manager
snmp-server enable traps frame-relay
snmp-server enable traps frame-relay
subif
snmp-server enable traps hsrp
snmp-server enable traps ipmobile
snmp-server enable traps ipmulticast
snmp-server enable traps mpls ldp
snmp-server enable traps mpls
traffic-eng
snmp-server enable traps mpls vpn
snmp-server enable traps msdp
snmp-server enable traps mvpn
snmp-server enable traps ospf state-
change
snmp-server enable traps ospf errors
snmp-server enable traps ospf
retransmit
snmp-server enable traps ospf lsa
```

```
snmp-server enable traps ospf cisco-
specific state-change nssa-trans-
change
snmp-server enable traps ospf cisco-
specific state-change shamlink
interface-old
snmp-server enable traps ospf cisco-
specific state-change shamlink
neighbor
snmp-server enable traps ospf cisco-
specific errors
snmp-server enable traps ospf cisco-
specific retransmit
snmp-server enable traps ospf cisco-
specific lsa
snmp-server enable traps pim
neighbor-change rp-mapping-change
invalid-pim-message
snmp-server enable traps pppoe
snmp-server enable traps cpu
threshold
snmp-server enable traps rsvp
snmp-server enable traps rtr
snmp-server enable traps syslog
snmp-server enable traps l2tun
session
snmp-server enable traps vsimaster
snmp-server enable traps vtp
snmp-server enable traps isakmp
policy add
snmp-server enable traps isakmp
policy delete
snmp-server enable traps isakmp
tunnel start
snmp-server enable traps isakmp
tunnel stop
snmp-server enable traps ipsec
cryptomap add
snmp-server enable traps ipsec
cryptomap delete
snmp-server enable traps ipsec
cryptomap attach
snmp-server enable traps ipsec
cryptomap detach
snmp-server enable traps ipsec tunnel
start
snmp-server enable traps ipsec tunnel
stop
snmp-server enable traps ipsec too-
many-sas
snmp-server enable traps voice poor-
qov
snmp-server enable traps voice
fallback
snmp-server enable traps dnis
snmp-server host 192.168.100.2 public
!
!
!
!
control-plane
!
!
!
!
!
!
!
!
!
line con 0
line aux 0
line vty 0 4
 login
!
scheduler allocate 20000 1000
!
end
```

9.4 Konfigurácia smerovača R4

9.4.1 – R4 s LLQ

```
service timestamps debug datetime
msec
service timestamps log datetime msec
no service password-encryption
!
hostname R4
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
!
resource policy
!
ip subnet-zero
!
!
ip cef
no ip dhcp use vrf connected
!
!
no ip ips deny-action ips-interface
!
!
!
!
class-map match-all voice-signaling
 match ip dscp cs3
class-map match-all voice-traffic
 match ip dscp ef
!
!
policy-map voice-policy
 class voice-traffic
  priority 350
 class voice-signaling
  bandwidth 15
 class class-default
  fair-queue
!
!
interface FastEthernet0/0
 description "Backup monitor
interface"
 ip address 192.168.100.3
255.255.255.0
 duplex auto
 speed auto
!
interface FastEthernet0/1
 description "R4 fa0/1 link na R5"
 ip address 192.168.20.1
255.255.255.0
 duplex auto
 speed auto
 no shutdown
!
interface Serial0/0/0
 description "S0 link na R6"
 ip address 192.168.24.1
255.255.255.0
 no fair-queue
 clockrate 2000000
 no shutdown
!
interface Serial0/0/1
 description "WAN link na R3"
 ip address 192.168.12.2
255.255.255.0
 ip route-cache flow
 clockrate 2000000
 service-policy output voice-policy
 no shutdown
!
router rip
 version 2
 network 192.168.12.0
 network 192.168.20.0
 network 192.168.24.0
!
ip classless
!
ip flow-export source FastEthernet0/1
ip flow-export version 9
ip flow-export destination
192.168.100.4 2000
!
ip http server
no ip http secure-server
!
snmp-server community public RW
snmp-server community testsip RW
snmp-server enable traps snmp
authentication linkdown linkup
coldstart warmstart
snmp-server enable traps vrrp
snmp-server enable traps tty
snmp-server enable traps xgcp
snmp-server enable traps flash
insertion removal
snmp-server enable traps envmon
snmp-server enable traps icsudsu
snmp-server enable traps isdn call-
information
snmp-server enable traps isdn layer2
snmp-server enable traps isdn chan-
not-avail
snmp-server enable traps isdn ietf
snmp-server enable traps ds0-busyout
snmp-server enable traps ds1-loopback
snmp-server enable traps atm subif
snmp-server enable traps bgp
snmp-server enable traps bulkstat
collection transfer
snmp-server enable traps cnpd
snmp-server enable traps config-copy
snmp-server enable traps config
snmp-server enable traps dial
snmp-server enable traps dsp card-
status
snmp-server enable traps entity
snmp-server enable traps frame-relay
snmp-server enable traps frame-relay
subif
snmp-server enable traps hsrp
```

```
snmp-server enable traps ipmobile
snmp-server enable traps ipmulticast
snmp-server enable traps msdp
snmp-server enable traps ospf state-
change
snmp-server enable traps ospf errors
snmp-server enable traps ospf
retransmit
snmp-server enable traps ospf lsa
snmp-server enable traps ospf cisco-
specific state-change nssa-trans-
change
snmp-server enable traps ospf cisco-
specific state-change shamlink
interface-old
snmp-server enable traps ospf cisco-
specific state-change shamlink
neighbor
snmp-server enable traps ospf cisco-
specific errors
snmp-server enable traps ospf cisco-
specific retransmit
snmp-server enable traps ospf cisco-
specific lsa
!
!
control-plane
!
line con 0
line aux 0
line vty 0 4
 login
!
scheduler allocate 20000 1000
!
end
```

9.4.2 – R4 s LFI, LLQ a cRTP

```
service timestamps debug datetime
msec
service timestamps log datetime msec
no service password-encryption
!
hostname R4
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
!
resource policy
!
ip subnet-zero
!
!
ip cef
no ip dhcp use vrf connected
!
!
no ip ips deny-action ips-interface
!
!
!
!
class-map match-all voice-signaling
 match ip dscp cs3
class-map match-all voice-traffic
 match ip dscp ef
!
!
policy-map voice-policy
 class voice-traffic
  priority 1823
 class voice-signaling
  bandwidth 15
 class class-default
  fair-queue
!
!
interface Multilink1
 ip address 192.168.12.2
255.255.255.0
 ip tcp header-compression iphc-
format
 service-policy output voice-policy
 no cdp enable
 no ip direct broadcast
 ppp multilink
 ppp multilink fragment-delay 20
 ppp multilink interleave
 multilink-group 1
 ip rtp header-compression iphc-
format
 max-reserved-bandwidth 100
!
!
interface FastEthernet0/0
 description "Backup monitor
interface"
 ip address 192.168.100.3
255.255.255.0
 duplex auto
 speed auto
!
interface FastEthernet0/1
 description "R4 fa0/1 link na R5"
 ip address 192.168.20.1
255.255.255.0
 duplex auto
 speed auto
 no shutdown
!
interface Serial0/0/0
 description "S0 link na R6"
 ip address 192.168.24.1
255.255.255.0
 no fair-queue
 clockrate 2000000
 no shutdown
!
interface Serial0/0/1
 description "WAN link na R3"
 bandwidth 2000
 clockrate 2000000
 description "link na r4 s LFI"
 no ip address
 encapsulation ppp
 ppp multilink
 multilink group 1
 no fair queue
 no shutdown
!
router rip
 version 2
 network 192.168.12.0
 network 192.168.20.0
 network 192.168.24.0
!
ip classless
!
ip flow-export source FastEthernet0/1
ip flow-export version 9
ip flow-export destination
192.168.100.4 2000
!
ip http server
no ip http secure-server
!
snmp-server community public RW
snmp-server community testsip RW
snmp-server enable traps snmp
authentication linkdown linkup
coldstart warmstart
snmp-server enable traps vrrp
snmp-server enable traps tty
snmp-server enable traps xgcp
snmp-server enable traps flash
insertion removal
snmp-server enable traps envmon
snmp-server enable traps icsudsu
snmp-server enable traps isdn call-
information
snmp-server enable traps isdn layer2
snmp-server enable traps isdn chan-
not-avail
snmp-server enable traps isdn ietf
snmp-server enable traps ds0-busyout
snmp-server enable traps ds1-loopback
snmp-server enable traps atm subif
snmp-server enable traps bgp
```

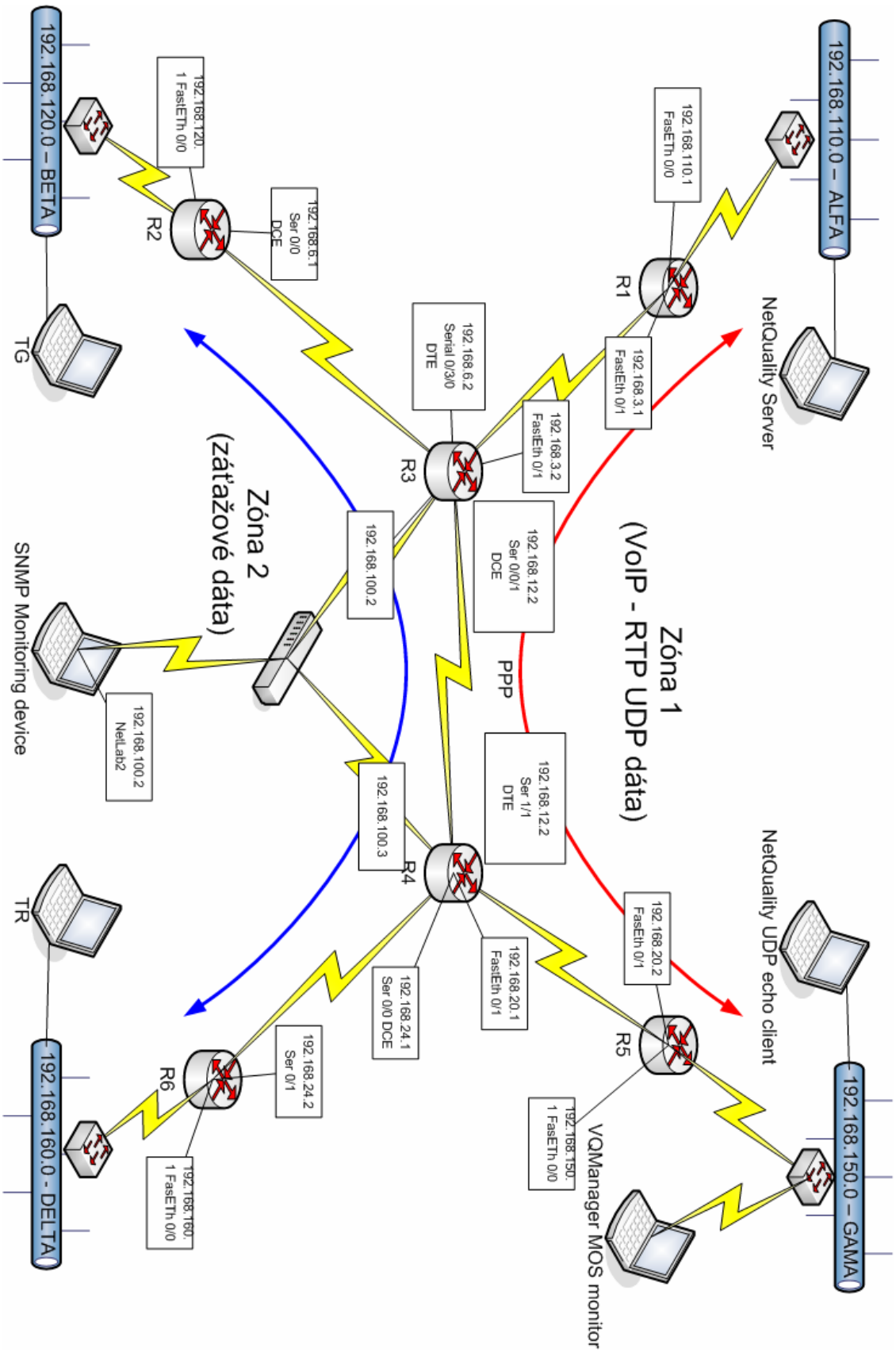
```
snmp-server enable traps bulkstat
collection transfer
snmp-server enable traps cnpd
snmp-server enable traps config-copy
snmp-server enable traps config
snmp-server enable traps dial
snmp-server enable traps dsp card-
status
snmp-server enable traps entity
snmp-server enable traps frame-relay
snmp-server enable traps frame-relay
subif
snmp-server enable traps hsrp
snmp-server enable traps ipmobile
snmp-server enable traps ipmulticast
snmp-server enable traps msdp
snmp-server enable traps ospf state-
change
snmp-server enable traps ospf errors
snmp-server enable traps ospf
retransmit
snmp-server enable traps ospf lsa
snmp-server enable traps ospf cisco-
specific state-change nssa-trans-
change
snmp-server enable traps ospf cisco-
specific state-change shamlink
interface-old
snmp-server enable traps ospf cisco-
specific state-change shamlink
neighbor
snmp-server enable traps ospf cisco-
specific errors
snmp-server enable traps ospf cisco-
specific retransmit
snmp-server enable traps ospf cisco-
specific lsa
!
!
control-plane
!
line con 0
line aux 0
line vty 0 4
 login
!
scheduler allocate 20000 1000
!
end
```

9.4.3 – R4 s IP RTP Priority

```
service timestamps debug datetime
msec
service timestamps log datetime msec
no service password-encryption
!
hostname R4
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
!
resource policy
!
ip subnet-zero
!
!
ip cef
no ip dhcp use vrf connected
!
!
no ip ips deny-action ips-interface
!
!
!
!
!
!
!
interface FastEthernet0/0
 description "Backup monitor
interface"
 ip address 192.168.100.3
255.255.255.0
 duplex auto
 speed auto
!
interface FastEthernet0/1
 description "R4 fa0/1 link na R5"
 ip address 192.168.20.1
255.255.255.0
 duplex auto
 speed auto
 no shutdown
!
interface Serial0/0/0
 description "S0 link na R6"
 ip address 192.168.24.1
255.255.255.0
 no fair-queue
 clockrate 2000000
 no shutdown
!
interface Serial0/0/1
 ip rtp priority 8000 16000 350
 description "WAN link na R3"
 ip address 192.168.12.2
255.255.255.0
 ip route-cache flow
 clockrate 384000
 fair-queue
 no shutdown
!
router rip
 version 2
 network 192.168.12.0
 network 192.168.20.0
 network 192.168.24.0
!
ip classless
!
ip flow-export source FastEthernet0/1
ip flow-export version 9
ip flow-export destination
192.168.100.4 2000
!
ip http server
no ip http secure-server
!
snmp-server community public RW
snmp-server community testsip RW
snmp-server enable traps snmp
authentication linkdown linkup
coldstart warmstart
snmp-server enable traps vrrp
snmp-server enable traps tty
snmp-server enable traps xgcp
snmp-server enable traps flash
insertion removal
snmp-server enable traps envmon
snmp-server enable traps icsudsu
snmp-server enable traps isdn call-
information
snmp-server enable traps isdn layer2
snmp-server enable traps isdn chan-
not-avail
snmp-server enable traps isdn ietf
snmp-server enable traps ds0-busyout
snmp-server enable traps ds1-loopback
snmp-server enable traps atm subif
snmp-server enable traps bgp
snmp-server enable traps bulkstat
collection transfer
snmp-server enable traps cnpd
snmp-server enable traps config-copy
snmp-server enable traps config
snmp-server enable traps dial
snmp-server enable traps dsp card-
status
snmp-server enable traps entity
snmp-server enable traps frame-relay
snmp-server enable traps frame-relay
subif
snmp-server enable traps hsrp
snmp-server enable traps ipmobile
snmp-server enable traps ipmulticast
snmp-server enable traps msdp
snmp-server enable traps ospf state-
change
snmp-server enable traps ospf errors
snmp-server enable traps ospf
retransmit
snmp-server enable traps ospf lsa
snmp-server enable traps ospf cisco-
specific state-change nssa-trans-
change
snmp-server enable traps ospf cisco-
specific state-change shamlink
interface-old
snmp-server enable traps ospf cisco-
specific state-change shamlink
neighbor
```

```
snmp-server enable traps ospf cisco-
specific errors
snmp-server enable traps ospf cisco-
specific retransmit
snmp-server enable traps ospf cisco-
specific lsa
!
!
control-plane
!
line con 0
line aux 0
line vty 0 4
 login
!
scheduler allocate 20000 1000
!
end
```

9.5 Schéma testovacej siete



Použitá Literatúra

[1] Cisco,VoIP over PPP Links with Quality of Service, www.cisco.com

[2] Dr. Peter J. Welcher, Quality of Service for Voice Over IP, 2003

[3] Syngress, Administering Cisco QoS in IP networks, 2002

[4] Minacom, MOS Technology Brief – Mean Opinion Score Algorithms for speech quality evaluation, 2007

[5] William C. Hardy, VoIP Service Quality – Measuring and evaluating packet switched voice, 2003

[6] Cisco, Voice over IP Fundamentals, 2000

[7] Cisco, QoS for Voice solution Guide, 2001

[8] Cisco, IP Traffic shaping and Policing, 2002

[9] Cisco , IP QoS queuing mechanisms, 2001

[10] www.voiptroubleshooter.com

[11] www.wikipedia.org

[12] www.voip-info.org