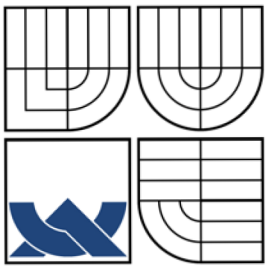
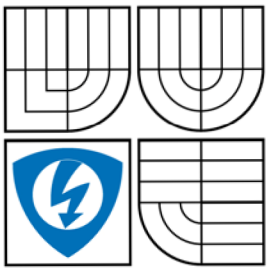


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

IMPLEMENTACE QOS V PŘÍSTUPOVÉ SÍTI

QOS IMPLEMENTATION IN ACCESS NETWORK

BAKALÁŘSKÁ PRÁCE
BACHELOR´S THESIS

AUTOR PRÁCE — VÁCLAV HAUSER
AUTHOR

VEDOUCÍ PRÁCE — doc. Ing. MILOSLAV FILKA, CSc.
SUPERVISOR

BRNO 2009

Anotace

Má bakalářská práce se zabývá problematikou implementace kvality služeb v přístupové síti. Nejdříve podrobně rozebírám samotný pojem kvality služeb a její vlastnosti. Dále čtenáře seznamuji s jednotlivými implementačními modely. Nejprve s integrovanými službami, které jsou založeny na rezervačním protokolu RSVP. Úkolem tohoto protokolu je vytvořit na dané trase požadovanou rezervaci pomocí zpráv RESV a PATH. Druhým implementačním modelem jsou diferencované služby. Zde není žádny rezervační protokol, kvalita služeb je zde zajištěna pomocí klasifikace jednotlivých datových toků. Podle toho, jak jsou pakety oklasifikovány, jim je přidělena priorita, podle které je s nimi dále v síti náležitě nakládáno. V další části práce popisuji příkazy pro samotnou implementaci různých druhů konfigurací obou implementačních modelů kvality služeb. Jedná se o konfiguraci RSVP protokolu, MQC, AutoQoS a MLS QoS. V závěrečné kapitole uvádím příklady těchto konfigurací.

Abstract

My bachelor thesis deals with the implementation of quality of service in the access network. My bachelor thesis describes at first the concept of quality of service and its characteristics. In addition, it makes the reader familiar with various implementations models. At first with integrated services, which are based on the RSVP reservation protocol. The task of this protocol is to establish the route through the reserve RESV and PATH messages. The second implementation model contains the differentiated services. There is no reservation protocol, quality of service is ensured by using the classification of individual data flows. Accordingly, as packets are classified, they are assigned according to a priority, which is crucial for the packets travel through the network. Next part of this thesis describes the implementation of any different types of configuration of two implementation models of quality of services. There is configuration of RSVP protocol, MQC, AutoQos and MLS QoS. Final chapter describes the configurations examples.

Klíčová slova (Key words): QoS, IntServ, RSVP, DiffServ, DSCP

Bibliografická citace:
HAUSER, V. Implementace QOS v přístupové síti. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009. 60 s. Vedoucí bakalářské práce doc. Ing. Miloslav Filka, CSc.

Prohlášení

Prohlašuji, že svou bakalářskou práci na téma Implementace QoS v přístupové síti jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením tohoto projektu jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne ..... **.....**

podpis autora

Obsah:

Seznam obrázků:

Seznam tabulek:

Úvod

Tématem mé bakalářské práce je Implementace QoS v přístupové síti. Toto téma bylo zadáno brněnskou firmou Maxprogres a mělo být vypracováváno v její spolupráci. V následujícím textu se tedy budu nejprve podrobněji zabývat samotným významem pojmu kvalita služeb, jejími základními parametry a vlivem na datový tok. Následně popíši základní implementačními modely QoS, které se dnes využívají. V první řadě tedy integrované služby (IntServ), které jsou historicky starším modelem, a dále služby diferencované, jinak řečeno rozlišované (DiffServ), které přišly jako reakce na zjištěné chyby svého předchůdce. U každého implementačního modelu nejprve vysvětlím jeho základní princip, popíši jeho komponenty, výhody a nevýhody. Dále se budu zabývat samotnou implementací QoS na síťové prvky, tedy rozeberu a popíši jednotlivé konfigurační příkazy ke každému modelu. Na závěr uvedu příklady jednotlivých konfigurací i s jejich popisem.

1 Co je QoS

1.1 Zavedení pojmu QoS

Již v počátcích internetu byla dána základní pravidla pro přenos, která byla následující [1]:

- žádnému provozu nebude odmítnut přístup do sítě
- se všemi provozy se bude zacházet stejně
- jedinou garancí pro provoz je, že data budou přenesena tím nejlepším způsobem (Best-Effort) v závislosti na dostupných prostředcích. Díky čemuž nedochází k umělému vytváření zpoždění nebo umělému zvyšování ztrátovosti paketů.

Negativní důsledky těchto pravidel se začaly projevovat po masovém rozšíření internetu v komerční sféře. Se stále rostoucím objemem přenášených dat se zvyšují nároky na mechanismy pro alokování síťových prostředků a na potřebnou šířku pásma. S takto rostoucími požadavky mají sítě stále větší a větší problémy. Nejvíce se tento problém projevuje u stále více se rozšiřujících multimediálních služeb. Běžné datové přenosy jsou charakteristické svými proměnnými nároky na šířku pásma a na spolehlivost spojení. Kdežto multimediální přenosy kladou větší nároky na konstantní šířku pásma, na malé zpoždění a je u nich všeobecně nastavena částečná tolerance ztrátovosti paketů, tudíž se jedná o spojení nespolehlivé. Jedná se o takzvané real-time systémy, tedy systémy pracující v reálném čase, kterými jsou například Voice-over-IP, Video-on-Demand, IP-TV atd. [18], [4] Vzhledem k tomu, že je dnešní internet založen na protokolu TCP/IP, který nedokáže s real-time datovými toky pracovat tak, jak je požadováno, byl zaveden pojem kvalita služeb (Quality of Services = QoS).

Pojem kvalita služeb v podstatě znamená, že daná síť je schopna rozeznat jednotlivé typy datových přenosů a pracovat s nimi odlišně podle stanovených požadavků. Například přiřazením různých priorit různým aplikacím, uživatelům nebo datovým tokům. Další možností je garance hodnot parametrů, které jsou stanoveny ještě před samotným vysíláním. [1], [7], [18]

Dále může být QoS definována jako souhrn přímo měřitelných parametrů spojení. [19]

1.2 Parametry QoS

Ke konkrétním parametrům QoS patří ztrátovost paketů, zpoždění, kolísání zpoždění a šířka pásma. [1], [4], [7], [8], [19] Nyní se podíváme na jednotlivé parametry podrobněji:

a) ztrátovost paketů (packet loss) – jedná se o poměr paketů, které nedorazí k příjemci, ku všem odeslaným paketům v určitém časovém období. U aplikací, které využívají protokol TCP, dojde při ztrátě paketů k jejich opětovnému poslání, což způsobí zpoždění, ale pro aplikace nepracující v reálném čase to není příliš velký problém. U aplikací pracujících v reálném čase je to právě naopak. Využívají protokol UDP, tedy takzvaný nespolehlivý přenos. Při ztrátě paketu nedojde k jeho opětovnému poslání, tudíž vznikají například mezery při konverzaci, které ovšem většinou dokáže kodek celkem úspěšně zamaskovat. Pokud není ztrátovost paketů příliš vysoká realtimovým aplikacím to nevadí, na rozdíl od zpoždění, které by vzniklo při opravování paketů. Vysoká ztrátovost po sobě jdoucích paketů by však měla za následek rapidní zhoršení kvality hovoru. Ztráta paketů může nastat z důvodu zahlcení síťových prvků na komunikační trase, například vyčerpáním kapacity vyrovnávacích pamětí, přetížením procesoru, dále pak třeba chybovostí spojů nebo kompletním výpadkem spoje s následným procesem hledání cesty v síti. Dalším faktorem, který mohou ovlivnit ztrátovost, je například vliv vnějšího rušení, které může porušit jeden nebo více bitů v paketu. Dále pak kolize na úrovni linkové vrstvy, či pozdržení paketu během přenosu na dobu delší než je nastaven časovač na přijímací straně. Tento paket je pak považován za ztracený. Například u VoIP je při náhodném rozložení ztráty paketů považováno za horní hranici ztrátovost 1 %. [1], [4], [7], [8], [19]

b) zpoždění (delay) – jinak nazývané také jednocestné zpoždění je vlastně čas, který zabere datům cesta z vysílacího zařízení do přijímacího zařízení. U hlasových služeb je zpoždění pro člověka zaznamenatelné pokud překročí hodnotu 150 ms, pokud převyšuje 200 ms je kvalita přenášeného hlasu již velice špatná. Celkové zpoždění je dáno součtem jednotlivých zpoždění aktivních a pasivních prvků sítě. Různé druhy zpoždění [7]:

- *zpoždění kódováním a paketizací* – sem patří doba, která je potřeba například na sestavení paketu a na sestavení aplikačního rámce
- *zpoždění při přenosu* – neboli doba šíření signálu médiem, je to funkce přenosové linky a je dána konečnou rychlostí šíření signálu po přenosové cestě
- *zpoždění ve frontě na odbavení* – jedná se o zpoždění způsobené čekáním paketu ve frontě aktivních prvků v přenosové cestě, tato hodnota se může dynamicky měnit
- *zpoždění při přepínání v síti* – toto zpoždění se může také dynamicky měnit, například instalací výkonnějšího směrovače

c) kolísání zpoždění (jitter) – jinak nazývané také rozptyl zpoždění nebo variace zpoždění. Je to rozdíl mezi celkovým zpožděním dvou paketů z téhož datového toku. V ideálním případě by měla být obě zpoždění stejně velká, tudíž by byl jitter nulový, ale ve skutečnosti se hodnoty zpoždění různých paketů liší, některé pakety tedy přijdou dříve a některé později. Jako řešení se například u VoIP využívá takzvaného *jitter bufferu*. Tento buffer udržuje pakety na přijímací straně, což umožňuje použít pakety, které jsou přijmuty mimo pořadí. Nevýhodou však je, že se tímto do komunikace zavádí další hodnota zpoždění. [1], [4], [7], [8], [19]

d) šířka pásma (bandwidth) – je to míra kmitočtového rozsahu. Její základní jednotkou je 1 Hz. Obecně platí, že čím větší šířku pásma v přenosovém kanálu můžeme využít, tím lze dosáhnout větší přenosové rychlosti. [1], [4], [7], [8], [19]

1.3 Vliv parametrů QoS na kvalitu

Různé druhy služeb mají různé nároky na nastavení hodnot parametrů sítě. V tabulce 1.1 jsou uvedeny všeobecné údaje o nárocích sítě bez ohledu na druh služby.

Tab. 1.1: Hodnoty síťových parametrů [7]

Parametr sítě	Dobrá	Akceptovatelná	Nevyhovující
Zpoždění	0 - 150 ms	150 - 300 ms	300+ ms
Jitter	0 - 20 ms	20 - 50 ms	50+ ms
Ztrátovost	0 - 0,5 %	0,5 - 1,5 %	1,5+ %

Když však půjdeme více do detailu a rozdělíme si jednotlivé služby, je vidět, jak se jejich nároky odlišují (viz tabulka 1.2). Jak vidíme tak například u přenosu hlasu je kladen hlavní důraz na dodržení stanovených hodnot zpoždění a kolísání zpoždění, aby byla zajištěna potřebná kvalita a nebyla komunikace znehodnocena. Na druhou stranu třeba u přenosu souborů není na zpoždění, ani na kolísání zpoždění kladen takový důraz, protože je pro nás důležité, aby se soubor přenesl kompletní a v pořádku i za cenu většího zpoždění.

Tab. 1.2: Citlivost různých druhů komunikace na parametry sítě [1]

Služby	Citlivost na			
	ztrátovost paketů	**zpoždění**	**kolísání zpoždění**	**šířku pásma**
Hlas	střední	vysoká	vysoká	velmi nízká
Elektronický obchod	vysoká	vysoká	nízká	nízká
E-Mail	vysoká	nízká	nízká	nízká
Telnet	vysoká	střední	nízká	nízká
Občasné prohlížení webu	střední	střední	nízká	nízká
Časté prohlížení webu	vysoká	vysoká	nízká	střední
FTP	střední	nízká	nízká	vysoká
Videokonference	střední	vysoká	vysoká	vysoká
Skupinové vysílání	vysoká	vysoká	vysoká	vysoká

2 Modely pro zajištění QoS v IP sítích

2.1 Integrované služby (IntServ)

Tento model byl první, který měl za úkol zajistit požadovanou kvalitu služeb v IP sítích. Jeho počátek sahá do roku 1994, kdy byl oficiálně definován [11]. Princip IntServ je následující. Aplikace si před samotným vysíláním definuje určité přenosové parametry, které bude vyžadovat po celou dobu spojení. Po vyslání tohoto požadavku do sítě, se na každém síťovém prvku ověří, zda je síť schopna tyto požadavky poskytnout. Pokud je odpověď kladná, na každém síťovém prvku dojde k rezervaci požadovaných prostředků, nejčastěji to bývají požadavky na zaručení určité šířky pásma, velikost vyrovnávací paměti atd. Pokud je však odpověď záporná mohou nastat dvě situace. Buď aplikace sníží své nároky, nebo se rozhodne přenos ukončit. [4], [7], [11], [18]

2.1.1 Implementační rámec IntServ

Implementační rámec IntServ se skládá ze 4 hlavních částí, které musí být implementovány ve všech směrovačích a hostitelích, tudíž ve všech odesílatelích a příjemcích QoS. Zde si tyto části uvedeme: [4], [7], [11]

a) plánovač paketů (packet scheduler) – stará se o plánování odesílání různých proudů paketů s využitím front, časovačů, priorit a dalších mechanizmů. Tento plánovač musí být implementován tam, kde jsou pakety zařazovány do front. Pakety různých služeb jsou dříve rozřazeny do různých tříd, podle nichž jsou dále roztříděny do různých front. Plánovač se dále zabývá odesíláním jednotlivých paketů z těchto front podle předepsaných algoritmů.

b) kontrola přístupu (admission control) – tato kontrola musí být spuštěna na každém uzlu sítě. Jedná se o prověření, zda je daný síťový prvek schopný zajistit požadované přenosové parametry, aniž by tím ovlivnil již existující rezervace.

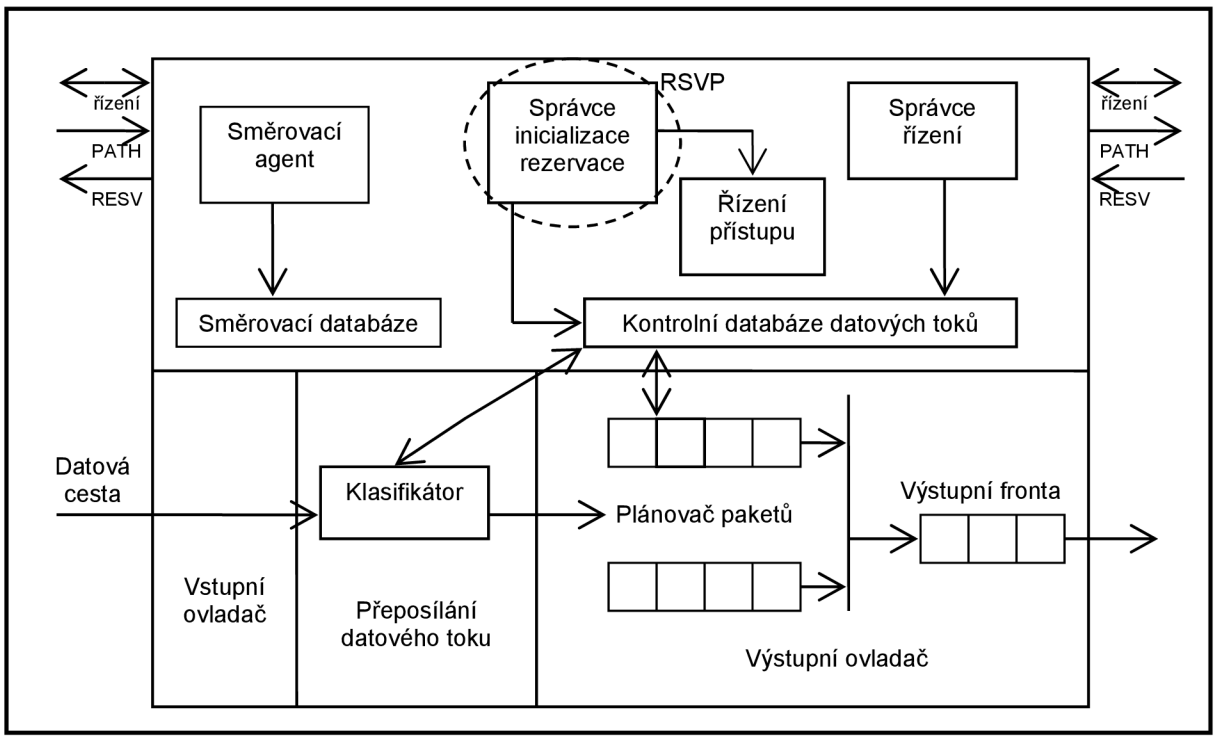
c) klasifikátor (classifier) – slouží k rozřazení jednotlivých paketů do různých tříd. Toto se děje hlavně kvůli efektivnímu řízení datových toků. Podle přidělené třídy od klasifikátoru jsou dále pakety rozřazeny

plánovačem do různých front (viz a). Do jaké třídy bude paket přiřazen se řídí například zdrojovou IP adresou, cílovou IP adresou, číslem portu atd.

d) protokol pro rezervaci prostředků (resource reservation protocol) – jedná se o takzvaný RSVP protokol. Je potřebný pro rezervaci prostředků, pro vytvoření a udržování stavů v koncových zařízeních a ve všech potřebných síťových uzlech na trase, kde má být provoz rezervován. [12] Budu se mu ještě podrobněji věnovat dále.

Následující obrázek (obr. 2.1) ukazuje umístění jednotlivých výše popsaných komponentů IntServ modelu ve směrovači a graficky znázorňuje jejich práci při zpracovávání paketů.

Po příchodu RSVP zprávy do směrovače, je tato zpráva přijata mechanizmem nazvaným řízení provozu, ten určí, zda je dané zařízení schopno zajistit požadovanou kvalitu služeb. Dvě hlavní součásti tohoto mechanizmu jsou klasifikátor (viz c) a plánovač paketů (viz a). Toto je však pouze první kritérium, které musí být splněno. Druhým kritériem je podstoupení takzvané policy kontrole.



Obr. 2.1: Umístění komponent IntServ modelu ve směrovači [7]

To znamená, že dojde k prověření uživatele, zda je vůbec oprávněn k vytvoření rezervace. Pokud jsou splněna obě tyto kritéria, dojde k nastavení klasifikátoru a plánovače paketů, čímž je zajištěna potřebná kvalita služeb. Naopak, pokud alespoň jedno kritérium splněno není, dojde k zamítnutí rezervace a RSVP odešle žadateli chybovou zprávu.

2.1.2 Protokol RSVP

Jedná se o signalizační protokol, který jak již bylo řečeno výše, slouží pro rezervaci prostředků v síti Internet a tím k získání určité kvality služeb pro jednotlivé datové toky. Směrovače ho využívají k doručení žádostí na kvalitu služeb ke všem uzlům na trase a k vytvoření a udržování stavů nutných pro poskytnutí kvality služeb. Z toho plyne, že rezervace musí být schválena a zajištěna u všech uzlů na trase. Pokud se na této trase vyskytují uzly, které nepodporují technologii IntServ, jsou přes ně zprávy RSVP protokolu protunelovány, zapouzdřením do IP paketů a dále předány na další směrovač. Další důležitou věcí je, že musí dojít k zajištění rezervace pro oba směry zvlášť, což je důvodem velké zátěže síťových prvků.

Pro řízení sítě využívá protokol RSVP následující strategie [1], [8], [12]:

- udržování stavu propojení
- hlídání a úprava přenosu
- předcházení zahlcení
- management předcházení nebo odstranění zahlcení
- mechanismus sledování výkonnosti linky

Kategorie aplikací, mezi kterými model IntServ rozlišuje [1], [8]:

- **elastické aplikace** – u těchto aplikací nejsou kladeny požadavky na omezení zpoždění nebo kapacitu spojení. Patří sem například e-mail, http protokol, atd.
- **reat-time tolerant aplikace (RTT)** – jedná se o aplikace, které tolerují občasnou ztrátu paketů, ale vyžadují omezení maximálního zpoždění. Patří sem například video aplikace, které využívají takzvané bufferování, což umožňuje skrytí ztráty paketů.

- **real-time intolerant aplikace (RTI)** – aplikace, které vyžadují minimální odezvu, rozptyl zpoždění a mají přísné požadavky na QoS. Například videokonference.

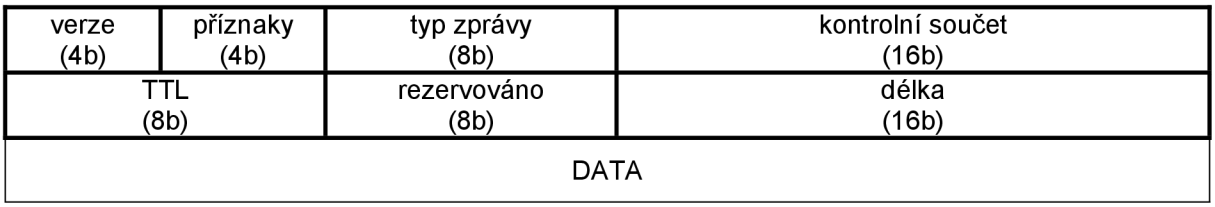
K zajištění obsluhy těchto aplikací slouží RSVP protokolu například následující třídy služeb (Class of Services) [1], [7], [8]:

- **zaručená služba (garanteed service)** – tato služba je určena pro RTI aplikace. Poskytuje záruku šířky pásma a hranice zpoždění. Důležitou vlastností je, že aplikace mohou snížit zpoždění zvýšením požadavků na šířku pásma.
- **služba s řízenou zátěží (controlled load service)** – služba je určena pro RTT aplikace. Zaručuje průměrné zpoždění.

2.1.3 Zprávy protokolu RSVP

Existuje několik druhů RSVP zpráv, jejichž plný výpis je v tabulce 2.1. Nejdůležitější zprávy pro funkčnost RSVP protokolu jsou zprávy PATH a RESV. Nyní si popíšeme podrobněji jejich funkci. Informace o potřebné rezervaci je od žadatele vyslána formou PATH zprávy. Tato zpráva musí být doručena a zpracována každým síťovým uzlem na požadované trase, což způsobí nastavení takzvaného PATH stavu na každém síťovém uzlu. Až se PATH zpráva dostane ke konečnému příjemci, ten odešle opačným směrem RESV zprávu, která má za úkol vytvořit rezervační stavy v každém uzlu na trase. Při nastavení PATH stavu byla v každém uzlu uložena adresa předchozího uzlu, aby byla zajištěna zpětná trasa pro RESV zprávu. Pokud RESV zpráva dorazí ke svému příjemci (tedy žadateli rezervace) je rezervace úspěšně ustanovena. Pokud však rezervace není možná nebo ji příjemce služby zamítne, vygeneruje se chybová zpráva a odešle se k odesílateli služby. [4], [7], [8], [12], [18]

RSVP zpráva je zabalena v IP paketu a ten v MAC rámci. Formát MAC rámce závisí na použité přenosové technologii. Konkrétní složení RSVP zprávy je vyobrazeno na obrázku 2.2. [7], [12]

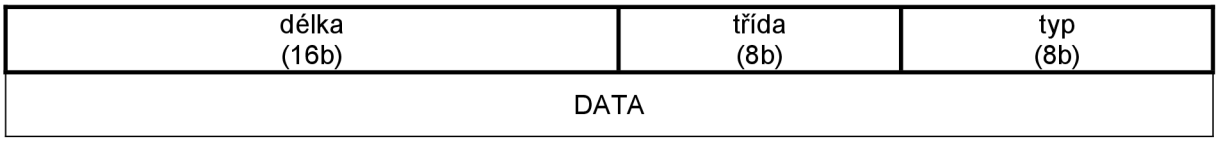


Obr. 2.2: Formát RSVP zprávy [7], [12]

- verze (version) (4 bity) – obsahuje číslo verze protokolu
- příznaky (flags) (4 bity) – 0x01 - 0x08: rezervovány
- typ zprávy (msg type) (8 bitů) – obsahuje informace o typu zprávy (viz tabulka 2.1)
- kontrolní součet (RSVP checksum) (16 bitů) – pokud je hodnota pole nulová, žádný kontrolní součet nebyl přenesen
- TTL (TimeToLive) (8 bitů) – hodnota významově shodná s IP TTL hodnotou
- délka (RSVP length) (16 bitů) – nese hodnotu celkové délky RSVP zprávy včetně hlavičky a objektů proměnné délky
- DATA – mají proměnnou délku a mohou se skládat z jednoho nebo více objektů (viz obrázek 2.3)

Tab. 2.1: Typy RSVP zpráv

číslo	typ zprávy
1	Path
2	Resv
3	PathErr
4	ResvErr
5	PathTear
6	ResvTear
7	ResvConf



Obr. 2.3: Formát RSVP objektu [7], [12]

- délka (length) (16 bitů) – nese hodnotu celkové délky objektu v bytech, hodnota musí být násobkem čtyř
- třída (class-num) (8 bitů) – identifikuje třídu objektu, každá tato třída má vlastní název. Příklady některých tříd jsou uvedeny dále:

a) NULL – může se objevit kdekoliv ve sledu jednotlivých objektů a jeho obsah je přijímačem ignorován
b) SESSION – obsahuje cílovou IP adresu a definuje specifickou relaci pro objekty, které ho následují, je vyžadován v každé RSVP zprávě
c) RSVP_HOP – nese IP adresu předchozího skoku (PHOP) nebo skoku následujícího (NHOP)
d) TIME_VALUES – obsahuje hodnoty obnovovacích dob zpráv, je vyžadováno v každé PATH a RESV zprávě
e) STYLE – definuje rezervační styl, vyžadováno v RESV zprávě
f) FLOWSPEC – definuje požadované QoS, v RESV zprávě
g) FILTER_SPEC – filtr, který definuje pakety přijímané QoS, v RESV zprávě
h) SENDER TEMPLATE – obsahuje IP adresu odesílatele a případně další informace pro jeho identifikaci, v PATH zprávě
i) SENDER_TSPEC – definuje množství dopravy, kterou generuje odesílatel, v PATH zprávě
j) ADSPEC – shrnutí QoS schopností, v PATH zprávě
k) ERROR_SPEC – specifikace chyby ve zprávách PathErr, ResvErr nebo potvrzení v ResvConfirm zprávě
l) POLICY_DATA – doposud ne zcela specifikovaná třída objektu, nese data, která jsou předávána ke kontrolnímu součtu modulu policy
m) INTEGRITY – nese autentifikační a ověřovací data
n) RESV_CONFIRM – obsahuje IP adresu kam se pošle rezervační potvrzení

- typ (C-type) (8 bitů) – obsahuje typ objektu, pro každou třídu je tento typ jedinečný

Dále podrobněji proberu základní druhy RSVP zpráv, a to zprávy PATH, RESV, Teardown a Chybové zprávy.

- **PATH zpráva** – tato zpráva je odesílána zdrojem, který chce vytvořit rezervaci pro určitou službu. Na cestě je zpracována každým směrovačem, načež je ve směrovači nastaven takzvaný PATH stav. PATH stav obsahuje IP adresu předchozího směrovače, aby bylo

zajištěno správné zpětné směrování pro RESV zprávu, zpět k odesílateli služby. Zpráva PATH se skládá z následujících objektů:
INTEGRITY, SESSION, RSVP_HOP, TIME_VALUES, POLICY_DATA, SENDER_TEMPLATE, SENDER_TSPEC, ADSPEC

Objekt INTEGRITY musí následovat jako první po hlavičce RSVP zprávy, na dalším pořadí objektů nezáleží. Dále popíši, co se stane s jednotlivými objekty před odesláním PATH zprávy do sítě. Do objektu SESSION bude vložena adresa cíle. Objekt RSVP_HOP obsahuje adresu následujícího skoku a zdroj do něj uloží také svou adresu. Dále je nastavena časová hodnota v objektu TIME_VALUES, tato hodnota udává, za jak dlouho se vygeneruje nová PATH zpráva, doporučená hodnota je 30 s. Podle této hodnoty je vypočten i čas, po jehož uplynutí bude zrušen PATH stav ve směrovači, pokud nepřijde nová PATH zpráva. Výpočet je proveden podle následujícího vzorce: $L = (k + 0{,}5) \cdot 1{,}5 \cdot R$, kde R představuje hodnotu z objektu TIME_VALUES a pro k je doporučená hodnota 3. Do objektu SENDER_TEMPLATE bude vložena IP adresa prvku, který požaduje QoS, aby jeho datový tok byl odlišen od ostatních toků v relaci. Kvantitativní vlastnosti služby jsou popsány v SENDER_TSPEC objektu a do posledního objektu, ADSPEC, jsou sbírány vlastnosti datové cesty a informace o podpoře dané služby. Poté je zpráva odeslána. Když zpráva dorazí do směrovače, upraví se objekt ADSPEC podle vlastností směrovače a vytvoří se PATH stav. Do PATH stavu jsou uloženy hodnoty z objektů SESSION, SENDER_TSPEC, RSVP_HOP a SENDER_TEMPLATE. Dále je změněna hodnota objektu RSVP_HOP a zpráva putuje dál. [1], [4], [7], [12]

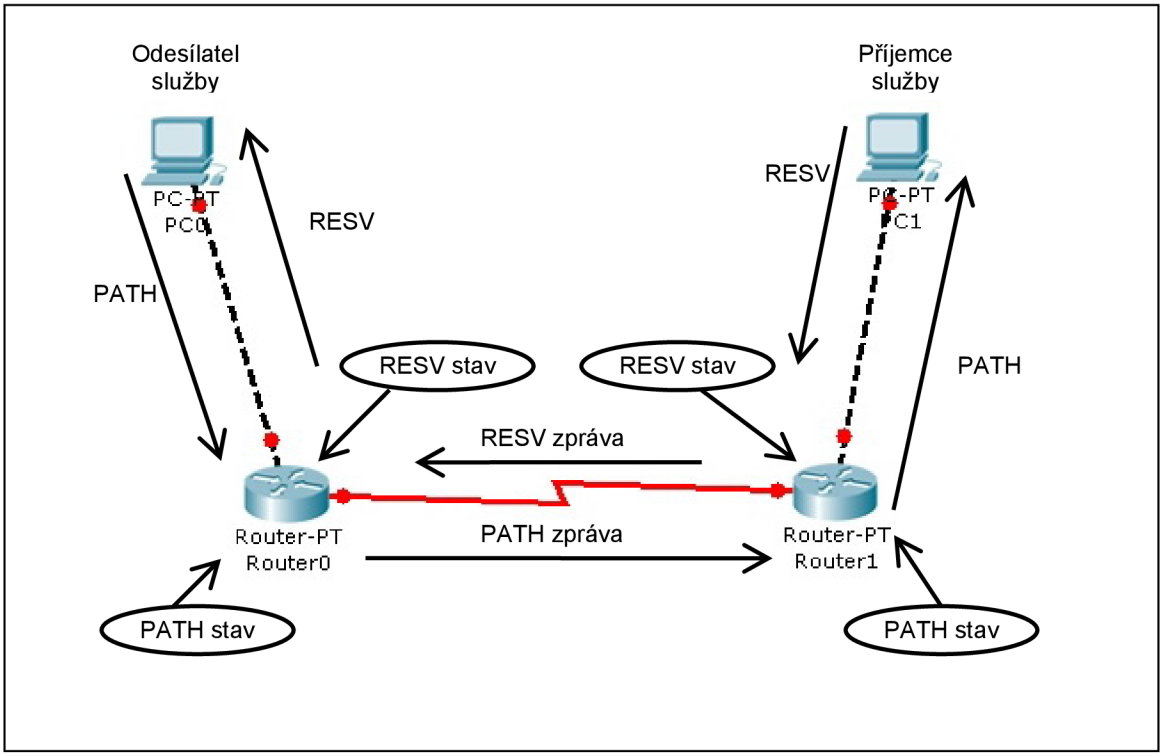
- **RESV zpráva** – zpráva RESV slouží jako odpověď na zprávu PATH. Nejdříve musí příjemce PATH zprávy vyhodnotit, zda danou službu přijme. Pokud se rozhodne přijmout službu a vytvořit pro ni rezervaci, odešle zpět rezervační zprávu RESV. Ta prochází zpět stejnou cestou, jako přišla zpráva PATH a postupně se snaží v jednotlivých uzlech rezervovat prostředky, jak bude popsáno dále. Zpráva RESV se skládá z následujících deseti objektů: INTEGRITY, SESSION, RSVP_HOP,

TIME_VALUES, RESV_CONFIRM, SCOPE, POLICY_DATA, STYLE, FLOWSPEC, FILTER_SPEC

Do objektu SESSION, je vložena stejná adresa jako byla v tomto objektu obsažena ve zprávě PATH, což je adresa příjemce služby. Do objektu RSVP_HOP je uložena adresa následujícího skoku v síti, zde je opět použita hodnota z PATH zprávy. Objekt RESV_CONFIRM nese adresu zařízení, které RESV zprávu vygenerovalo. Objekt STYLE obsahuje druh rezervačního stylu. Parametry požadované služby jsou obsaženy v objektu FLOWSPEC a do FILTER_SPEC se uloží adresa žadatele QoS. Pro směrovače jsou nejdůležitější objekty FLOWSPEC a FILTER_SPEC, jejichž hodnoty slouží k nastavení parametrů klasifikátoru a plánovače paketů. Po přijetí zprávy směrovačem se musí rozhodnout, zda má směrovač dostatek zdrojů k poskytnutí rezervace. Toto rozhodnutí může být učiněno na základě stávajících rezervací ve směrovači, nebo podle aktuálního stavu výstupního rozhraní. Pokud se směrovač rozhodne rezervaci přijmout, nastaví se RESV stav, do kterého jsou uloženy hodnoty objektů SESSION, RESV_CONFIRM, STYLE, FLOWSPEC a FILTER_SPEC. Na závěr je změněna hodnota v objektu RSVP_HOP a RESV zpráva pokračuje v cestě. Pokud se buď příjemce služby nebo některý ze směrovačů rozhodne rezervaci nepřijmout, je vygenerována chybová zpráva a odeslána odesílateli služby. [1], [4], [7], [12]

- **Teardown zprávy** – tyto zprávy mohou být dvou typů, jednak *PathTear*, nebo *ResvTear* a slouží k odstranění PATH nebo RESV stavu ve směrovačích. Odeslání této zprávy koncovými zařízeními po ukončení aplikace, není nezbytně nutné, ale doporučené. Dalším případem, kdy mohou být tyto zprávy rozeslány, je například situace, kdy v časovém limitu, který je dán hodnotou v objektu TIME_VALUES, nedorazí nová PATH nebo RESV zpráva. [1], [4], [7], [12]
- **Chybové zprávy** – opět existují dva typy chybových zpráv, a to *PathErr* a *ResvErr*. Zpráva PathErr je vygenerována pokud nastane chyba v poslání PATH zprávy. Je do ní uložen druh chyby a adresa směrovače, na kterém se chyba vyskytla, a je odeslána zpět odesílateli služby. Zpráva ResvErr je vygenerována v případě, že nastala chyba v posílání

RESV zprávy, a je opět odeslána odesílateli služby, který může následně změnit své požadavky a znovu se pokusit o rezervaci prostředků. [1], [4], [7], [12]



Obr. 2.4: Úspěšná výměna RSVP zpráv

2.1.4 Nevýhody a využití Integrovaných služeb

IntServ je vhodný spíše pro menší sítě. Více se hodí pro implementaci například do přístupových sítí, důvodem jsou stále se zvyšující nároky na paměť a výkon směrovačů s rostoucí velikostí sítě. Další nevýhodou je požadovaná podpora RSVP protokolu u všech směrovačů, přes které procházejí IP pakety datových toků. Pokud je IntServ využíván, tak v menších sítích a nebo na okraj rozlehlejší sítě, ale spíše je dnes již vytlačet diferencovanými službami. Například v podnikových sítích je využíván při zajišťování QoS pro přenos hlasu, kdy před ustanovením komunikace probíhá kontrola přijetí spojení pomocí protokolu RSVP. Další nevýhodou je orientace RSVP protokolu na přijímač, který musí být hlavním iniciátorem rezervace zdrojů. RSVP protokol selže, pokud vysílač nechce ustanovit datový tok s definovanou QoS. [7], [8], [11], [18]

2.2 Diferencované služby (DiffServ)

Jako reakce na výše uvedené nevýhody integrovaných služeb vznikl pokročilejší a principielně jednodušší model pro zajištění QoS, a to diferencované služby. V tomto modelu jsou data vstupující do sítě klasifikována podle jejich nároků na síť a poté přiřazena do různých druhů tříd (Class of Services), přičemž každá třída má přidělen vlastní režim agregace. Podle přiřazené třídy jsou následně data upravena. Identifikace třídy, do které jsou data přiřazena, se nejčastěji provádí pomocí hlavičky IP protokolu. [4], [8], [18] Zde záleží na verzi IP protokolu, pokud se jedná o verzi 4, je použito pole Typ služby (ToS), nebo pole Třída provozu (TC), pokud se jedná o IP protokol verze 6. Hlavičky paketů obou verzí IP protokolů jsou uvedeny na obrázku 2.5 a 2.6. [9], [13], [17]

<table>
<tr><td>Verze IP
(4b)</td><td>Délka záhlaví
(4b)</td><td>Typ služby
(8b)</td><td colspan="2">Celková délka IP datagramu
(16b)</td></tr>
<tr><td colspan="3">Identifikace IP datagramu
(16b)</td><td>Příznaky
(3b)</td><td>Offset fragmentu
(13b)</td></tr>
<tr><td colspan="2">TTL
(8b)</td><td>Protokol vyšší vrstvy
(8b)</td><td colspan="2">CRC
(16b)</td></tr>
<tr><td colspan="5">IP adresa odesílatele
(32b)</td></tr>
<tr><td colspan="5">IP adresa příjemce
(32b)</td></tr>
<tr><td colspan="5">Volitelné položky záhlaví</td></tr>
<tr><td colspan="5">DATA</td></tr>
</table>

Obr. 2.5: Hlavička paketu IPv4 [9]

<table>
<tr><td>Verze IP
(4b)</td><td>Třída dat
(8b)</td><td colspan="3">Identifikace toku dat
(20b)</td></tr>
<tr><td colspan="2">Délka dat
(16b)</td><td>Další hlavička
(8b)</td><td colspan="2">Počet hopů
(8b)</td></tr>
<tr><td colspan="5">Zdrojová IP adresa
(128b)</td></tr>
<tr><td colspan="5">Cílová IP adresa
(128b)</td></tr>
<tr><td colspan="5">Volitelné položky</td></tr>
<tr><td colspan="5">DATA</td></tr>
</table>

Obr. 2.6: Hlavička paketu IPv6 [13]

2.2.1 DiffServ Code Point (DSCP)

Pro další vysvětlování je nutné si objasnit význam základních pojmů. Pomocí služby DS Code Point se identifikuje režim agregace, pro kód této služby se využívají výše zmíněná pole IP hlavičky. Pole je tímto přejmenováno na DS pole. Z celkových osmi bitů je pro kód DSCP využito pouze šest bitů, viz obrázky 2.7 a 2.8. Z toho plyne, že může být vyjádřeno celkově 64 různých hodnot DSCP. [14], [17] Tyto hodnoty jsou rozděleny do tří skupin: [1], [14]

a) 32 standardizovaných Code Pointů
b) 16 Code Pointů dostupných pro experimentální účely nebo pro lokální využití
c) 16 Code Pointů dostupných pro výzkum a lokální využití, ale pokud se první kategorie plně obsadí, mohou být tyto také standardizovány

Priorita (3b)	D (1b)	T (1b)	R (1b)	Pro budoucí využití (2b)

Obr. 2.7: Pole Typ služby v IP datagramu [10]

- priorita (precedence) (3 bity) – určuje prioritu dané služby
- D (delay) (1b) – zpoždění
- T (throughput) (1b) – propustnost
- R (reliability) (1b) – spolehlivost přenosu

DSCP (6b)	Pro budoucí využití (2b)

Obr. 2.8: DS pole v IP datagramu [14]

2.2.2 Per Hop Behaviour (PHB)

Ke každému DSCP je přidruženo chování paketů ve vnitřní síti, takzvané Per Hop Behaviour. PHB tedy představuje navenek pozorovatelný režim zpracovávání a zasílání paketů v jednotlivých síťových uzlech. PHB poskytuje prostředky, díky nimž může uzel přidělit zdroje k režimu agregace. Jednotlivé PHB jsou seskupovány do skupin, které jsou tvořeny na základě podobných vlastností PHB, jako jsou například požadavky na šířku pásma atd. Existují standardizované PHB, které jsou určeny k širokému užívání. Tyto PHB mají

doporučenou hodnotu DSCP, která je určena v hlavičce paketu. Každý využívaný DSCP musí být namapován na určité PHB, pokud není, jsou data buď přiřazena ke standardnímu PHB (službě Best Effort) nebo zahozena. Dále si uvedeme příklady třech navržených PHB, které v současné době existují: [4], [7], [16]

a) **standardní a class selector PHB** – standardní PHB je využíváno pro pakety, které nejsou namapovány na žádný standardizovaný ani místní PHB a je jim přiřazena služba Best Effort. Ve srovnání se všemi ostatními PHB má standardní přiřazenu nejnižší prioritu, to znamená, že má přidělenou bitovou hodnotu DSCP 000000. Priorita je dána hodnotou prioritních bitů v DSCP poli IP paketu. Podpora těchto prioritních bitů je aplikována pomocí Class Selector PHB. Všechny prioritní kombinace jsou uvedeny v tabulce 2.2.

Tab. 2.2: Priority v DSCP poli [7]

hodnota DSCP	priorita
000000	Best Effort
001000	7. nejvyšší priorita
010000	6. nejvyšší priorita
011000	5. nejvyšší priorita
100000	4. nejvyšší priorita
101000	3. nejvyšší priorita
110000	2. nejvyšší priorita
111000	nejvyšší priorita

b) **zajištění předávání (assured forwarding)** – Tato služba slouží k přenosu paketů garantovanou rychlostí. Poskytuje čtyři třídy úrovní záruk a zdrojů, jako například prostor vyrovnávací paměti, šířka pásma, pro přijaté pakety atd. Vedle tohoto rozdělení jsou dále pakety v každé třídě rozděleny do dalších tří prioritních skupin pro zahazování v případě zahlcení síťového prvku. Pakety s vyšší drop prioritou budou zahozeny s větší pravděpodobností než pakety s nižší drop prioritou. Podle toho v jaké třídě a drop prioritní skupině se paket nachází, je mu přiřazena hodnota AF třídy. Soupis AF tříd a jejich DSCP hodnot je v tabulce 2.3. První tři bity DSCP hodnoty odpovídají IP prioritě, takzvanému IP Precedence. Následující dva bity určují pravděpodobnost zahození a poslední bit je 0.

Tab. 2.3: Hodnoty AF tříd [7], [16]

Drop priorita	Třída 1 (AF1x)	Třída 2 (AF2x)	Třída 3 (AF3x)	Třída 4 (AF4x)
Nízká (AFx1)	001010	010010	011010	100010
Střední (AFx2)	001100	010100	011100	100100
Vysoká (AFx3)	001110	010110	011110	100110

c) **urychlené předávání (expedited forwarding)** – Tato služba se také nazývá prémiovou. Poskytuje nástroje pro vytvoření nízké ztrátovosti, zpoždění a jitteru. Zajištění této služby je velmi složité a neefektivní, jelikož poskytování služby jakémukoliv toku znamená vytvoření virtuálního okruhu, čímž je způsobeno nižší využití síťových prostředků. Služba má doporučenou hodnotu DSCP rovnu 101100.

V tabulce 2.4 je souhrn nejčastěji využívaných PHB hodnot spolu s přiřazenými hodnotami CoS, IP priority a DSCP.

Tab. 2.4: Hodnoty CoS, IP Precedence, DSCP a PHB [2]

CoS	IPP	DSCP	PHB	typická aplikace
7	7			Rezervováno
6	6	48	CS6	Routing
5	5	46	EF	Hlas
5	5	34	AF41	Video konference
4	4	32	CS4	Streamované video
3	3	26	AF31	Mission critical data
3	3	24	CS3	Call Signaling
2	2	18	AF21	Transaction data
2	2	16	CS2	Network Management
1	1	10	AF11	Bulk data
1	1	8	CS1	Scavenger
0	0	0	0	Best effort data

2.2.3 Service Level Agreements a Traffic Conditioning Agreements

Pravidla, která si například zákazník dohodne s poskytovatelem a podle nichž se mají pakety chovat, se nazývají Service Level Agreements (SLA). Podmnožinou k SLA je dohoda o úpravě provozu Traffic Conditioning Agreements (TCA). TCA podrobně specifikuje způsob, jakým bude s daty zacházeno, aby bylo zaručeno domluvené SLA. TCA obsahuje klasifikační pravidla, dopravní profily, značení a pravidla pro formování datového toku. TCA dále obsahuje dvě součásti, a to nucenou TCA a strukturovanou TCA. Nucená

TCA slouží jako ochrana poskytovaných zdrojů v každé DiffServ úrovni služeb a strukturovaná TCA určuje přidanou per flow, která může být nabízena dodavatelem. [7], [8], [15] V tabulce 2.4 jsou příklady dvou úrovní služeb, které byly aplikovány na pakety se specifickou hodnotou DSCP.

Tab. 2.5: Příklady TCA [7]

DSCP	průměrná rychlost	úroveň služby
000000	100 kbps	Best Effort
000001	1 Mbps	Better Best Effort

Jak již bylo řečeno, TCA je základní součástí SLA. Další součásti SLA je například cenový a účtující mechanismus, ověřovací mechanismy, šifrovací služby a další. Vyjednávání SLA může být dvojího typu, jednak statické, což je současná norma a pak dynamické. Dynamické vyjednávání SLA je mnohem náročnější, vyvstávají při něm nejrůznější problémy, jako například problém s kompatibilitou klientských zařízení atd. [7], [15]

2.2.4 DiffServ domény

Jelikož je internetová síť nesmírně rozlehlá, je velice obtížné zajistit jednotné zpracování požadavků na QoS. Proto je síť rozdělena do menších oblastí se samostatnou správou. V případě diferencovaných služeb mluvíme o takzvaných DiffServ doménách. Tyto domény obsahují dva druhy směrovačů. Jednak jsou to vnitřní směrovače a jednak hraniční směrovače. Dále si rozebereme jejich funkce: [1], [8]

- **vnitřní směrovače (core router)** – zajišťují pouze vnitřní spojení diffserv domény, neprovádí žádnou klasifikaci, s pakety zachází podle jejich značek
- **hraniční směrovače (edge router)** – tyto směrovače se ještě dále dělí na další dvě skupiny a to:
    a) ingress směrovače – zajišťují značkování paketů
    b) egress směrovače – zajišťují odznačení paketů

Do DiffServ domény vstupují pakety z části sítě, která obsahuje ještě neoznačkované pakety. Vstupují přes ingress hraniční směrovače, kde jsou veškeré vstupní pakety klasifikovány (označkovány) a odeslány do DiffServ

domény. Klasifikace, například zavedení DS pole, může proběhnout na základě IP adresy odesílatele nebo adresáta, čísel portů, podle výsledků měření dynamických vlastností přicházejících dat apod. Dále jsou pakety přenášeny beze změny značky vnitřními směrovači až na okraj domény, kde z ní přes egress hraniční směrovače vystupují. Pokud spolu sousedí dvě DiffServ domény, pak hraniční směrovač na rozhraní dvou domén pracuje současně jako ingress pro jednu doménu a egress pro doménu druhou. Jelikož dvě domény nemusí mít shodná pravidla, většinou dochází na rozhraní těchto domén k rekvalifikaci paketů. [1], [8]

2.2.5 Hranice důvěryhodnosti

Tento pojem souvisí s předchozí kapitolou o DiffServ doménách. Hranice důvěryhodnosti je prostor, ve kterém se nachází zařízení, jehož označkování provozu je označeno za důvěryhodné. Z hlediska aplikace kvality služeb je nejlepší, pokud je hranice důvěryhodnosti co nejblíže koncovému zařízení, ideálně tedy v přístupové síti. [1]

Existují tři druhy zařízení z hlediska důvěryhodnosti: [1]

a) důvěryhodné zařízení (trusted endpoint) – disponuje schopnostmi pro značkování provozu odpovídající značkou. Další úlohou je přeznačkovávání provozu, který byl značkován nedůvěryhodným zařízením. Nejčastěji jsou jako důvěryhodná zařízení například IP telefony, servery, videokonferenční systémy atd.

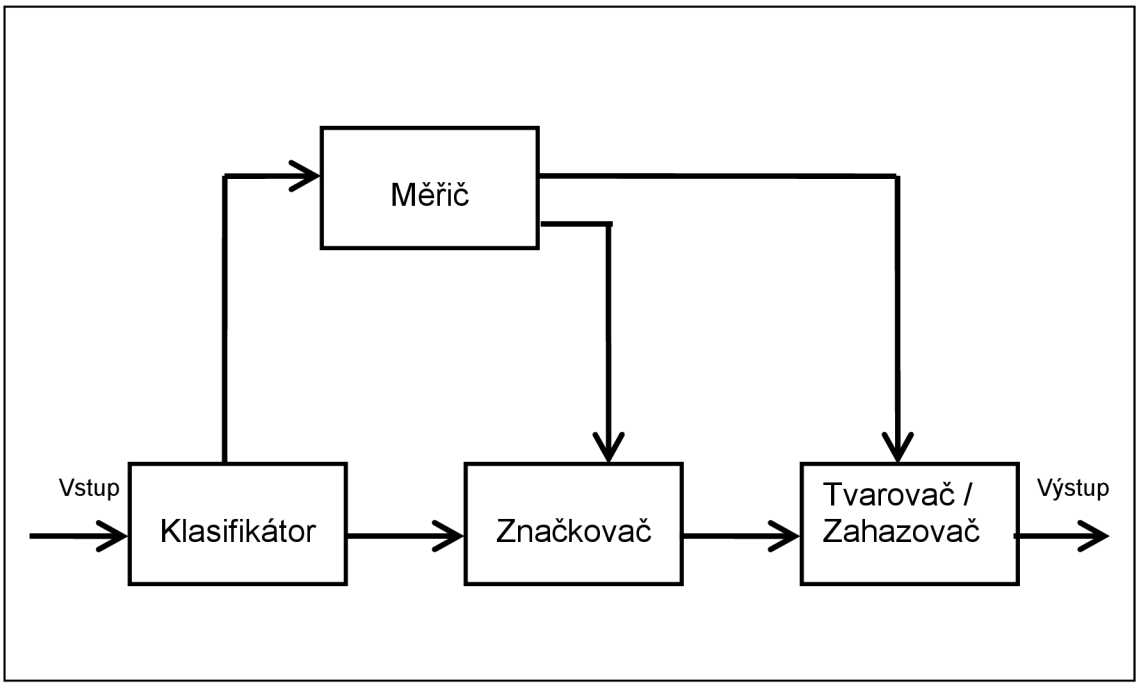
b) nedůvěryhodné zařízení (untrusted endpoint) – při označkování provozu tímto zařízením dojde při nejbližší příležitosti k jeho přeznačkování. Za nedůvěryhodná zařízení jsou nejčastěji označovány koncové počítače, z důvodu možnosti neoprávněného značkování.

c) podmíněně důvěryhodné zařízení (conditionally trusted endpoint) – takto jsou označována zařízení jako například IP telefony, které z důvodu mobility mohou měnit své umístění na jednotlivých portech směrovače. Tímto by mohl vyvstat problém, jelikož, jak již bylo zmíněno, PC není považováno za důvěryhodné zařízení. Proto byl firmou Cisco implementován protokol CDP (Cisco Discovery Protocol),

který je využíván pro sdílení informací o jiných přímo připojených zařízeních.

2.2.6 Základní komponenty modelu DiffServ

Data, která mají vstoupit do DiffServ domény, musejí být patřičně upravena, aby odpovídala pravidlům specifikovaným v TCA. Tato úprava je vykonávána v hraničních ingress směrovačích. V podstatě se jedná o převzetí vstupních paketů a jejich umístění do front v takovém pořadí, aby byly co nejlépe splněny požadavky na zasílání, a síťové zdroje by byly využity co nejlepším způsobem. Na obrázku 2.9 je zmiňovaná úprava znázorněna. [1], [4], [7], [15]



Obr. 2.9: Úprava provozu v modelu DiffServ [4]

Jak je vidět na obrázku, úprava provozu obsahuje čtyři základní komponenty, a to klasifikátor, měřič, značkovač a tvarovač/zahazovač, dále si je podrobněji popíšeme: [1], [4], [7], [15]

a) **klasifikátor (classifier)** – tato komponenta má na starosti vybírání paketů z datového toku, které se děje na základě obsahu hlavičky paketu. Využívány jsou dva druhy klasifikace, a to:

- *sdružené zacházení* (behavior aggregate) – klasifikace probíhá podle DSCP
- *vícepoložková klasifikace* (multi-field) – klasifikace probíhá podle kombinace jedné či více hodnot, a to například podle zdrojové adresy, cílové adresy, DS pole, ID protokolu, čísla zdrojového či cílového portu atd.

Také se může stát, že klasifikace již proběhla při generaci v dané aplikaci. V takové situaci je pak na směrovači, zda se rozhodne této klasifikaci důvěřovat či nikoliv. Pokud ne, dojde k reklasifikaci.

b) **měřič (meter)** – měřič má za úkol monitoring vlastností klasifikovaných datových toků a jejich porovnávání s dopravním profilem TCA. Zjištěné stavy datových toků měřič odesílá značkovači a tvarovači/zahazovači. V některých modelech úpravy provozu je tento blok označován jako *policing* a dokonce slučován s tvarovačem (*policing and shaping*).

c) **značkovač (marker)** – značkovač nastavuje DS pole paketů na určitý DSCP, podle kterého je přiřazen k danému režimu chování (PHB). Funkce značkovače závisí na stavu měřiče, může značkovat buď všechny příchozí pakety nebo pouze určité z nich.

d) **tvarovač (shaper)** – pomocí tvarovače jsou datové toky uváděny do souladu s dopravním profilem (TCA) pomocí zpožďování nebo urychlování paketů. Tuto schopnost tvarovači umožňuje jeho vyrovnávací paměť. Pokud dojde k přetečení paměti, některé pakety jsou vyřazeny.

e) **zahazovač (dropper)** – zahazovač má stejný účel jako tvarovač, tedy upravuje datový tok, aby odpovídal TCA, s tím rozdílem, že zahazovač provádí úpravy vyřazováním některých nebo všech paketů. Může být také implementován jako zvláštní druh tvarovače s nulovou velikostí vyrovnávací paměti.

2.2.7 Využití diferencovaných služeb

Na rozdíl od integrovaných služeb u služeb diferencovaných aplikace neoznamuje předem do sítě své požadavky na QoS, tudíž není nutné využívat

žádného rezervačního protokolu a zátěž pro síťové prvky je mnohem menší. Z tohoto důvodu jsou diferencované služby dnes mnohem více využívané než služby integrované. Více se hodí pro rozlehlejší sítě a pro jádra sítí. [8], [18]

3 Implementace QoS

3.1 Integrované služby (IntServ)

Jak jsem již zmínil výše, pro implementaci integrovaných služeb se využívá signalizačního protokolu RSVP. Při samotné implementaci RSVP protokolu musí být na daném síťovém prvku nejdříve nakonfigurován mechanismus zpracovávání paketů podle použitého typu služby. Při využití služby s řízenou zátěží se používá mechanismus předcházení zahlcení WRED (Weighted Random Early Detection) a při využití zaručené služby mechanismus správy zahlcení WFQ (Weighted Fair Queueing). Dále může RSVP využívat mechanismus LLQ (Low Latency Queueing). Dále blíže popíši uvedené mechanismy [2], [6]:

a) **WRED (Weighted Random Early Detection)** – předchází zahlcení tím, že selektivně zahazuje méně významné pakety. K rozlišování paketů využívá například IP priority, kdy doručuje více paketů s vyšší prioritou a větší množství paketů s nižší prioritou zahazuje. Dále preferuje datové toky RSVP protokolu před ostatními. Dokáže přimět TCP/IP protokol k opakovanému odeslání zahozených paketů se sníženou rychlostí, čímž se dá předejít takzvané globální synchronizaci. [2], [6]

b) **WFQ (Weighted Fair Queueing)** – dělí provoz do jednotlivých toků, mezi které spravedlivě rozděluje pásmo. V jednom toku jsou pakety se stejnou zdrojovou a cílovou adresou, stejným zdrojovým a cílovým TCP či UDP portem a protokolem. Opět jsou využity IP priority, toky s vyšší prioritou jsou účinněji zpracovány. Pro rozhraní s rychlostí E1 (2048 Mbps) je tento mechanismus nastaven jako defaultní. Pro zahazování paketů se využívá *Modified tail* drop systém, který má dva režimy. Jednak *early dropping*, který zahazuje dříve, než se fronta naplní, nebo *aggressive dropping*, který zahazuje až při naplnění fronty. [2], [6]

c) **CBWFQ (Class-based Weighted Fair Queueing)** – jedná se o mechanismus třídního WFQ, provoz je tedy klasifikován podle definic map tříd. Pro každou třídu je umožněno garantovat šířku pásma a využívá se samostatná fronta. Pro zahazování paketů se využívá *Tail Drop* systém, který se chová ke každému provozu stejně. Po zaplnění fronty se pakety

začnou zahazovat, dokud zahlcení nepomine. Případně se může nastavit mechanismus WRED. [2]

d) LLQ (Low Latency Queueing) – jedná se o nejpoužívanější mechanismus. Umožňuje vytvořit vysoce prioritní frontu s garantovanou šířkou pásma a nízkou latencí, na které je aplikován policing, aby v případě zahlcení nebyla překročena zadaná šířka pásma. Za normálních okolností ovšem může být daná hodnota překročena. Zvláště pro hlasový provoz umožňuje vytvoření striktně prioritní fronty do CBWFQ s redukcí jitteru. [6]

Dále RSVP protokol umožňuje spolupráci se službou COPS (Common Open Policy Service), která slouží ke správě RSVP. Vzájemná spolupráce může probíhat přes rozhraní Ethernet, T1 a HSSI. Služba COPS ke svému chodu využívá PDP (Policy Decision Points), což jsou servery s běžícím Cisco QoS Policy Managerem a PEP (Policy Enforcement Points), což jsou v podstatě klienti služby COPS, nejčastěji směrovače. [6]

3.1.1 Příkazy pro konfiguraci RSVP

Zvolím si konfiguraci RSVP spolu s LLQ, tudíž nejdříve představím příkazy pro konfiguraci fronty mechanismu WFQ. [6]

1) Aktivace WFQ na rozhraní:

- (interface) **fair-queue** [*práh-zahazování* [*d-fronty* [*r-fronty*]]]
  - *práh-zahazování* – (nepovinné) určuje maximální počet nových paketů v každé frontě (v rozsahu 16 – 4096), po dosažení této hodnoty se začnou další pakety zahazovat. Defaultně je nastaveno na hodnotu 64.
  - *d-fronty* – (nepovinné) definuje počet dynamických front pro provoz best effort. Může nabývat hodnot 16, 32, 64, 128 256, 512, 1024, 2048, 4096.
  - *r-fronty* – (nepovinné) definuje počet front pro rezervovaný provoz, jako například pro RSVP. Může být v rozsahu 0 – 1000.

2) (Nepovinné) Využití skupiny QoS nebo DWFQ založené na ToS

a) Zapnutí DWFQ

- (interface) **fair-queue** { **qos-group** | **tos** }

- *qos-group* – určuje, že provoz se bude do front řadit podle lokálně platných QoS skupin
- *tos* – určuje, že provoz se bude do front řadit podle typu služby

b) Nastavení přenosové kapacity pro skupiny

- (interface) **fair-queue** { **qos-group** *skupina* | **tos** *tos* } **weight** *procento*
  - *weight* – slouží k přidělení procentuální hodnoty přenosové kapacity každé skupině QoS nebo každému číslu ToS při zaplnění přenosové kapacity.

c) (Nepovinné) Nastavení celkového počtu paketů ve frontách DWFQ

- (interface) **fair-queue aggregate-limit** *paketů*
  - *aggregate-limit* – nastavení celkového počtu paketů, které je možné uložit do front před začátkem zahazování

d) (Nepovinné) Nastavení celkového počtu paketů v jedné frontě

- (interface) **fair-queue individual-limit** *paketů*
  - *individual-limit* – nastavení maximálního počtu paketů ve frontě jednoho toku

3) (Nepovinné) Nastavení striktně prioritní fronty pro provoz RTP protokolu

- (interface) **ip rtp priority** *počáteční-port počet-portů kapacita*
  - *ip rtp priority* – nastavuje identifikaci RTP paketů podle rozsahu UDP portů a zaručenou přenosovou kapacitu pásma

Po nastavení WFQ mechanismu již můžeme přejít k příkazům pro konfiguraci RSVP protokolu. [5], [6]

4) Konfigurace RSVP s LLQ

a) Zapnutí RSVP se striktními prioritními frontami

- (global) **ip rsvp pq-profile** [**voice-like** | *r‘* [*b‘* [*p-to-r‘* | **ignore-peak-value**]]]
  - *voice-like* – (nepovinné) nastaví pro hlasové toky typický profil prioritních front. Hlasové toky jsou směřovány do striktně prioritní fronty mechanismu WFQ

- *r‘* – (nepovinné) nastavení hodnoty maximální přenosové rychlosti (v bps)
- *b‘* – (nepovinné) nastavení hodnoty maximální špičky (v bajtech)
- *p-to-r‘* – (nepovinné) nastavení hodnoty maximálního poměru špičkové rychlosti k průměrné rychlosti (v procentech)
- *ignore-peak-value* – (nepovinné) když je uveden tento příkaz nedochází k výpočtu poměru špičkové rychlosti k průměrné

b) Zapnutí RSVP na daném rozhraní

- (interface) **ip rsvp bandwidth** [*kbps-na-rozhraní* [*kbps-na-tok*]]
  - *kbps-na-rozhraní* – (nepovinné) definuje maximální objem rezervace přenosu na daném rozhraní (v kbps)
  - *kbps-na-tok* – (nepovinné) definuje maximální rychlost rezervovanou pro jeden tok (v kbps)

c) (Nepovinné) Nastavení faktoru reakce na špičku

- (interface) **ip rsvp burst policing** [*faktor*]
  - *faktor* – (nepovinné) definuje špičkový faktor (v procentech), což je hodnota, která říká jak přísně mají být uhlazeny přenosové špičky. Čím nižší je hodnota, tím přísnější uhlazení.

d) (Nepovinné) Zapnutí funkce RSVP proxy pro systémy bez podpory RSVP

- (global) **ip rsvp reservation** *ip-adresa-cíle ip-adresa-zdroje* { **tcp** | **udp** | *id-protokolu* } *d-port s-port ip-adresa-dalšího-hopu rozhraní-dalšího-hopu* { **ff** | **se** | **fw** } { **rate** | **load** } *rychlost špička*
  - *ip rsvp reservation* – nastavení vysílání a příjmu zpráv RSVP RESV
- (global) **ip rsvp sender** *ip-adresa-cíle ip-adresa-zdroje* { **tcp** | **udp** | *id-protokolu* } *d-port s-port ip-adresa-předchozího-hopu rozhraní-předchozího-hopu rychlost špička*
  - *ip rsvp sender* – nastavení vysílání a příjmu zpráv RSVP PATH
  - *ip-adresa-cíle* – adresa zamýšleného příjemce provozu
  - *ip-adresa-zdroje* – adresa zdroje provozu

- *tcp* | *udp* | *id-protokolu* – definuje druh protokolu přenášeného rezervací. Identifikátor protokolu může být v rozsahu 0 – 255.
- *d-port* – definuje cílový port příjemce. Pokud je zadána hodnota 0, předpokládá se, že buď aplikace nepoužívá čísla portů, nebo že se rezervace vztahuje na všechny cílové porty.
- *s-port* – definuje zdrojový port odesílatele
- *rozhraní-dalšího* | *předchozího-hopu* – definuje druh rozhraní, jako například ethernet, loopback, null nebo seriál
- *ff* | *se* | *wf* – definice typu rezervace
  - *ff (fixed filter)* – jediná rezervace pro každý zdroj
  - *se (shared explicit)* – jedna rezervace pro všechny zdroje, které se určují při vytváření rezervace
  - *wf (wildcard filter)* – jedna rezervace pro všechny zdroje, které se přidávají automaticky kdykoliv v průběhu rezervace
- *rate* | *load* – předmět rezervace
  - *rate* – zaručená rychlost
  - *load* – řízená zátěž
- *rychlost* – přenosová rychlost rezervace, může být až 75 % celkové přenosové kapacity rozhraní (v kbps)
- *špička* – maximální velikost špičky (v kB)

5) Nastavení služby COPS

a) Nastavení COPS serverů

- (global) **ip rsvp policy cops servers** *ip-serveru* [*ip-serveru …..*]
  - *ip serveru* – je možno zadat až osm COPS serverů. První uvedená IP adresa je brána jako primární a další jako záložní.

b) (Nepovinné) RVP hlášení pro servery

- (global) **ip rsvp policy cops report-all**
  - *report-all* – zařízení posílá na servery informace o veškerých rozhodnutích RSVP.
- (global) **ip rsvp policy cops minimal**
  - *minimal* – serverům se posílají pouze informace o zprávách RESV a PATH

6) Další užitečné příkazy

a) Omezení příjmu rezervací pouze od vybraných stanic

- (interface) **ip rsvp neighbors** *access-list-number*
  - *access-list-number* – číselné označení standardního (1 – 99) nebo rozšířeného (100 – 199) access listu.

b) Informace o RSVP na jednotlivých rozhraních

- (global) **show ip rsvp interface** [*type-number*]
  - *type-number* – (nepovinné) typ rozhraní s jeho číselným označením

c) Informace o zavedených filtrech a šířce pásma

- (global) **show ip rsvp installed** [*type-number*]
  - *type-number* – (nepovinné) typ rozhraní s jeho číselným označením

d) Zjištění současných RSVP sousedů

- (global) **show ip rsvp neighbor** [*type-number*]
  - *type-number* – (nepovinné) typ rozhraní s jeho číselným označením

e) Informace o RSVP požadavcích

- (global) **show ip rsvp request** [*type-number*]
  - *type-number* – (nepovinné) typ rozhraní s jeho číselným označením

f) Informace o rezervovaných tocích, které se nacházejí v databázi

- (global) **show ip rsvp reservation** [*type-number*]
  - *type-number* – (nepovinné) typ rozhraní s jeho číselným označením

g) Informace o vysílačích, které se nacházejí v databázi

- (global) **show ip rsvp sender** [*type-number*]
  - *type-number* – (nepovinné) typ rozhraní s jeho číselným označením

3.2 Diferencované služby (DiffServ)

3.2.1 Modular QoS Command Line Interface

Pro implementaci diferencovaných služeb na svých zařízeních zavedla firma Cisco implementační systém MQC (Modular QoS Command Line Interface). Jak název napovídá, jedná se o řádkové rozhraní QoS. MQC využívá k definování provozu mapy tříd (*class map*), které umožňují pružnou klasifikaci provozu. Mapy tříd se dále využívají v mapách politik (*policy map*), kterými se určí co se má s vybraným provozem dělat. A nakonec se mapy politik aplikují na jednotlivých rozhraních jako politiky služby (*service policy*). Mechanismy použitelné pomocí MQC [1], [2], [6]:

a) **marking** – označování paketů (nastavení DSCP, IP precedence atd.)
b) **shaping** – omezování provozu se zpožďováním (rozložením)
c) **policing** – omezování provozu zahazováním paketů
d) **WFQ** – garance šířky pásma
e) **LLQ** – garance šířky pásma a nízké čekací doby
f) **WRED** – garance šířky pásma s inteligentním zahazováním paketů

Dále blíže popíši jednotlivé kroky konfigurace MQC. [1], [2], [6]

1) **Definice mapy tříd (Traffic class definition)** – V tomto kroku se definují jednotlivé mapy tříd, čímž určíme pravidla pro rozřazování provozu na rozhraních do různých tříd. Často se také využívá ACL. Pokud jsou ACL přiřazena do mapy tříd, rozřazování se drží následujících pravidel:
   - při nalezení shody s pravidlem permit se použije přiřazená QoS akce
   - při nalezení shody s pravidlem deny se dané ACL přeskočí
   - při nenalezení shody s žádným pravidlem permit se nepoužije QoS ale Best Effort
   - když je definováno více ACL, hledání se zastaví po první shodě s pravidlem permit
2) **Definice mapy politik (Policy definition)** – Nyní je nutno již rozřazenému datovému toku zadat, jak se má dále chovat. Jednak se provádí *marking*, úprava šířky pásma, řeší se fronty atd.

3) **Aplikace mapy politik (Policy application)** – Aplikace mapy politik probíhá obdobně jako aplikace ACL. Nejdříve je třeba určit rozhraní, na kterém se má daná mapa uplatňovat a posléze ještě zda se týká příchozích či odchozích paketů. Dají se také vytvářet různé hierarchie politik, to znamená, že se mohou uvnitř jedné politiky aplikovat další.

3.2.2 Příkazy pro konfiguraci MQC

1) Klasifikace provozu přes mapy tříd

a) Definice mapy tříd

- (global) **class-map** [**match-all** | **match-any**] *název-mapy-tříd*
  - *match-all* – (nepovinné) paket je přiřazen do dané třídy pokud vyhovuje všem stanoveným podmínkám
  - *match-any* – (nepovinné) paket je přiřazen do dané třídy pokud vyhovuje libovolné podmínce
  - *název-mapy-tříd* – každá mapa tříd má přidělený libovolný název

b) Definice testovacích podmínek

- (class-map) **match any**
  - do takto zadefinované třídy budou spadat všechny pakety
- (class-map) **match class-map** *název-mapy-třídy*
  - tímto příkazem docílíme vnoření jedné mapy třídy do druhé
- (class-map) **match protocol** *protocol*
  - do takto zadefinované třídy budou spadat pakety, které obsahují zadaný protokol
- (class-map) **match access-group** [*číslo-acl* | **name** *název-acl*]
  - do takto zadefinované třídy budou spadat pakety, které projdou přes zvolené ACL. ACL volíme pomocí čísla nebo jména.
- (class-map) **match cos** *cos* [*cos cos cos*]
  - do takto zadefinované třídy budou spadat pakety, které mají nastavenou danou hodnotu CoS. Je možné zadat až čtyři hodnoty priority CoS.
- (class-map) **match ip precedence** *priorita* [*priorita priorita priorita*]

- do takto zadefinované třídy budou spadat pakety, které mají nastavenou danou prioritu protokolu IP. Je možné zadat až čtyři hodnoty priority.

- (class-map) **match ip dscp** *dscp* [*dscp dscp dscp dscp dscp dscp dscp dscp*]
  - do takto zadefinované třídy budou spadat pakety, které se shodují pouze s jednou zadanou hodnotou DSCP
- (class-map) **match qos-group** *skupina-qos*
  - do takto zadefinované třídy budou spadat pakety, které vyhovují dané skupině QoS. Tyto QoS skupiny jsou nastaveny ve směrovači a mají pouze lokální význam.
- (class-map) **match mpls experimental** *hodnota-mpls*
  - do takto zadefinované třídy budou spadat pakety, které mají nastavenou experimentální hodnotu MPLS (Multiprotocol Label Switching) na zadanou hodnotu. Jedná se o tříbitové pole v hlavičce MPLS.
- (class-map) **match ip rtp** *počáteční-port počet-dalších-portů*
  - do takto zadefinované třídy budou spadat pakety, které obsahují hodnotu UDP portu ze zvoleného rozsahu
- (class-map) **match source interface** *typ číslo*
  - do takto zadefinované třídy budou spadat pakety, které byly přijaty zvoleným rozhraním
- (class-map) **match source-address mac** *mac-adresa*
  - do takto zadefinované třídy budou spadat pakety, které přišly z nastavené zdrojové MAC adresy. Nelze použít na sériových rozhraních nebo u ATM.
- (class-map) **match destination-address mac** *mac-adresa*
  - do takto zadefinované třídy budou spadat pakety, které jsou určeny pro nastavenou cílovou MAC adresu

2) Nastavení politiky QoS

a) Definice mapy politiky

- (global) **policy-map** *název-politiky*

– *název-politiky* – každá mapa politiky má přidělený libovolný název

b) Přiřazení jedné nebo více tříd k politikám

- (pmap) **class** *název-třídy*
  - *název-třídy* – určuje třídu na jejíž provoz se bude aplikovat daná politika

c) (Nepovinné) Použití výchozí třídy

- (pmap) **class class-default**
  - *class-default* – do výchozí třídy spadá veškerý provoz, který neodpovídá žádné nadefinované třídě

d) (Nepovinné) Nastavení parametrů QoS paketů

- (pmap-class) **set fr-de**
  - *set fr-de* – tímto příkazem se nastaví příznak DE (Discard Eligibility) u Frame Relay, který není u paketů konvertovaných na rámce standardně nastaven. Nastavením tohoto příznaku určujeme paket jako vhodného kandidáta na zahození v případě zahlcení.
- (pmap-class) **set atm-clp**
  - *set atm-clp* – tímto příkazem se nastaví příznak CLP (Cell Loss Priority) u ATM, který je u paketů konvertovaných na buňky standardně nastaven na nulu. Pokud jeho hodnotu změníme na 1 určíme paket jako vhodného kandidáta na zahození v případě zahlcení.
- (pmap-class) **set cos** *cos*
  - *set cos* – tímto příkazem se nastaví hodnota CoS. Nastavuje se pouze u paketů, které se předávají v přepínaném prostředí.
  - *cos* – námi zvolená hodnota CoS v rozsahu 0 – 7 (0 = nízká, 7 = vysoká)
- (pmap-class) **set ip dscp** *dscp*
  - *dscp* – námi zvolená hodnota DSCP v rozsahu 0 – 63, přičemž se dají využít i hodnoty EF (urychlené předávání), AF11 (zaručené předávání třídy 11) atd.
- (pmap-class) **set ip precedence** *priorita*

– *priorita* – námi zvolená hodnota IP priority v rozsahu 0 – 7 (0 = nejnižší, 7 = nejvyšší)

- (pmap-class) **set mpls experimental** *hodnota-mpls*
  – *hodnota-mpls* – námi zvolená hodnota experimentálního pole v hlavičce MPLS. Může nabývat hodnot 0 – 7.
- (pmap-class) **set qos-group** *skupina-qos*
  – *skupina-qos* – námi zvolená skupina QoS v rozsahu 0 – 99.

e) (Nepovinné) Nastavení správy zahlcení

- (pmap-class) **bandwidth** { *kapacita* | **percent** *procento* }
  – *bandwidth* – nastavení WFQ
  – *kapacita* – přidělení určité kapacity linky dané třídě (v kbps)
  – *percent* – pokud není známa kapacita linky přiděluje se kapacita pomocí procentuálního vyjádření (v procentech)
- (pmap-class) **priority** { *kapacita* | **percent** *procento* } [*špička*]
  – *priority* – nastavení LLQ
  – *kapacita* – přidělení určité kapacity linky dané třídě (v kbps)
  – *percent* – pokud není známa kapacita linky přiděluje se kapacita pomocí procentuálního vyjádření (v procentech)
  – *špička* – (nepovinné) maximální velikost špičky
- (pmap-class) **fair-queue** [*počet-front*]
  – *fair-queue* – nastavení počtu dynamických front dostupných výchozí třídě (*class-default*) v rozsahu 16 – 4 096, přičemž hodnoty musí být mocniny dvou (výchozí nastavení je 16)
- (pmap-class) **queue-limit** *paketů*
  – *queue-limit* – stanovení maximálního počtu paketů ve frontě v rozsahu 1 – 64. Po dosažení maximální hodnoty se další pakety zahazují, dokud se fronta neuvolní.

f) (Nepovinné) Nastavení předcházení zahlcení

- (pmap-class) **random-detect** [**prec-based** | **dscp-based**]
  – *random-detect* – nastavení WRED
  – *prec-based* – (nepovinné) WRED se bude řídit hodnotou IP priority (výchozí nastavení)

- *dscp-based* – (nepovinné) WRED se bude řídit podle hodnoty DSCP

- (pmap-class) **random-detect** { **precedence** *priorita* | **dscp** *dscp* } *minimální-práh maximální-práh 1-z-N*
  - *priorita* | *dscp* – volba paketů, které se mají zahazovat
  - *minimální-práh* – po dosažení minimálního prahu se začnou zahazovat některé z paketů se shodnou hodnotou *priorita* | *dscp*
  - *maximální-práh* – po dosažení maximálního prahu se začnou zahazovat všechny pakety se shodnou hodnotou *priorita* | *dscp*
  - *1-z-N* – nastavení z kolika (*N*) paketů se má vždy jeden zahodit při dosažení minimálního prahu

g) (Nepovinné) Nastavení přenosové politiky

- (pmap-class) **police** *bps burst-normal burst-max* **conform-action** *akce* **exceed-action** *akce* [**violate-action** *akce*]
  - *bps* – nastavení průměrného provozu (v bps)
  - *burst-normal* – hodnota, o kterou může provoz narůst přes průměrnou hodnotu (v bajtech)
  - *burst-max* – špičková hodnota nárůstu provozu (v bajtech)
  - *conform-action* – volíme akci, která se má provádět, pokud provoz dosahuje maximálně hodnoty *burst-normal*
  - *exceed-action* – volíme akci, která se má provádět pokud je provoz mezi hodnotami *burst-normal* a *burst-max*
  - *violate-action* – (nepovinné) volíme akce, která se má provádět pokud je provoz nad hodnotou *burst-max*. Pokud není tato akce zvolena, nemá hodnota *burst-max* žádný význam.
  - *akce* – můžeme zvolit následující akce:
    - **drop** – zahazování paketů
    - **set-prec-transmit** *nová-priorita* – paket je znovu zpracován se změněnou IP prioritou
    - **set-qos-transmit** *nová-qos* – paket je znovu zpracován se změněnou skupinou QoS
    - **set-dscp-transmit** *nová-dscp* – paket je znovu zpracován se změněnou hodnotou DSCP

- **transmit** – paket je odeslán

h) (Nepovinné) Nastavení třídního mechanismu GTS (Generic Traffic Shaping)

- (pmap-class) **shape** { **average** | **peak** } *cir* [*bc*] [*be*]
  - *shape* – nastavení mechanismu GTS, který omezuje odchozí provoz na zadanou rychlost, čímž dokáže předcházet zahlcení
  - *average* – nastavení průměrné rychlosti (v bps)
  - *peak* – nastavení špičkové rychlosti (v bps)
  - *cir* (*commited information rate*) – průměrná přenosová rychlost (v bps)
  - *bc* (*commited burst size*) – počet bytů, o které je povoleno jednorázově překročit průměrnou přenosovou rychlost
  - *be* (*excess burst size*) – povolený přenos nad rámec nastavené špičky

3) Přiřazení mapy politik k rozhraní

- (interface) **service-policy** { **input** | **output** } *název-mapy*
  - *input* | *output* – nastavení příchozího nebo odchozího směru na rozhraní kde bude politika uplatňována
  - *název-mapy* – název politiky, kterou chceme přiřadit

4) (Nepovinné) Nastavení striktně prioritní fronty pro hlasový přenos

- (interface) **ip rtp priority** *počáteční-port-rtp počet-portů kapacita*
  - *ip rtp priority* – nastavení striktně prioritní fronty, která je zpracovávána před ostatními frontami na rozhraní
  - *počáteční-port-rtp* – první UDP port z rozsahu
  - *počet-portů* – volba rozsahu UDP portů
  - *kapacita* – zaručená přenosová kapacita (v kbps)

5) Další užitečné příkazy

a) Informace o všech zadefinovaných mapách tříd

- (global) **show class-map**

b) Podrobné informace o zvolené mapě tříd

- (global) **show class-map** *název-mapy-tříd*

c) Informace o všech zadefinovaných mapách politik

- (global) **show policy-map**

d) Podrobné informace o zvolené třídě v dané mapě politik

- (global) **show policy-map** *název-mapy-politik* **class** *název-mapy-tříd*

e) Statistiky uplatnění politiky na daném rozhraní

- (global) **show policy-map interface** *název-rozhraní*

3.2.3 Auto-QoS

Stejně jako MQC je Auto-QoS určeno pro zařízení firmy Cisco. Jedná se o nejjednodušší a nejrychlejší nastavení QoS, ale je primárně určeno pouze pro hlasové služby, tedy pro VoIP provoz. Je použitelné pro Cisco IP telefony, Cisco SoftPhone a uplink porty, takzvané trunky. Po zapnutí Auto-QoS nejdříve dané zařízení provede diagnostiku sítě a následně podle toho nakonfiguruje QoS. Nevýhoda Auto-QoS je v tom, že automatickou konfigurací může dojít k přepsání dříve definované uživatelské konfigurace, proto je doporučeno Auto-QoS použít před eventuální uživatelskou konfigurací QoS. Nespornou výhodou je spolupráce s protokolem CDP, který slouží k detekci Cisco telefonů v síti. Pomocí Auto-QoS jsou pakety klasifikovány do následujících tří tříd [1], [2], [3]:

- voice rtp pakety
- voice signalizace
- ostatní

3.2.4 Příkazy pro konfiguraci Auto-QoS

1) Zapnutí Auto-QoS

- (interface) **auto qos voip** { **cisco-phone** | **cisco-softphone** | **trust** }
  - *cisco-phone* – nastavíme pokud je k zařízení připojen Cisco IP telefon
  - *cisco-softphone* - nastavíme pokud je k zařízení připojen Cisco SoftPhone
  - *trust* – dané rozhraní je připojeno k důvěryhodnému přepínači nebo směrovači, tudíž je příchozím paketům důvěřováno

2) Další užitečné příkazy

a) Kontrola vložených příkazů na daném rozhraní

- (global) **show auto qos interface** *název-rozhraní*

3.2.5 Multilayer Switch QoS

Multilayer Switche (MLS) jsou přepínače, které poskytují kromě služeb 2. vrstvy OSI modelu také některé služby vyšších vrstev. Mezi tyto služby patří například také QoS. V hierarchii sítě tvoří přepínače nejčastěji její přístupovou část, kde se provádí klasifikace a značkování. Při zpracovávání provozu využívají přepínače DSCP hodnoty, které odpovídají prioritě provozu, ale pro výběr výstupní fronty se využívají hodnoty CoS. Proto existují takzvané mapovací tabulky, které umožňují převod mezi DSCP a CoS hodnotami. Existují ještě další popisy mapovacích tabulek [2], [3]:

- **CoS-DSCP map** – slouží pro převod CoS na DSCP (defaultní nastavení viz tabulka 3.1)
- **DSCP-CoS map** – slouží pro převod DSCP na CoS (defaultní nastavení viz tabulka 3.2)
- **IP-Prec-DSCP map** – slouží pro převod IP priority na DSCP
- **DSCP-mutation** – tuto mapu použijeme, pokud spojujeme dvě domény, které využívají odlišné hodnoty DSCP. Na hranici sítí díky této mapě přepíšeme DSCP na naše hodnoty.
- **Policed-DSCP** – tato mapa se využívá při klasifikaci a značkování provozu ke přepsání DSCP na novou hodnotu

Tab. 3.1: Defaultní CoS-DSCP mapa [2]

CoS	0	1	2	3	4	5	6	7
DSCP	0	8	16	24	32	40	48	56

Tab. 3.2: Defaultní DSCP-CoS mapa [2]

DSCP	0-7	8-15	16-23	24-31	32-39	40-47	48-55	56-63
CoS	0	1	2	3	4	5	6	7

Ke správnému nastavení QoS na přepínači ještě dále patří nastavení vstupních a výstupních front. Dále popíši dva nejvyužívanější mechanismy. [2]

a) **WRR (Weighted Round Robin)** – plánovač front, který se využíval u starších druhů přepínačů. Mechanismus *Round Robin* pracuje následovně: z výstupních front se postupně při každém průchodu odebírají pakety a odesílají se. Jednotlivým frontám se také dají přidělovat takzvané *váhy*. Tím se dá ovlivnit kolik paketů se odebere z dané fronty během jednoho průchodu, čímž se ovlivňuje rychlost odbavování jednotlivých front.

b) **SRR (Shaped/Shared Round Robin)** – u novějších modelů byla technologie WRR změněna na technologii SRR. Vedle výstupní fronty se zde využívá i fronta vstupní. SRR může pracovat ve dvou módech, a to *Shared mode* a *Shaped mode*.

- *Shared mode* – tento mód garantuje šířku pásma pro frontu, ale současně pásmo neomezuje, takže zbývající šířka pásma může být využita ostatními frontami
- *Shaped mode* – tento mód garantuje šířku pásma pro frontu a současně pásmo i limituje na zadanou hodnotu

SRR využívá jako zahazovací mechanismus technologii *WTD* (*Weighted Tail Drop*) se třemi prahovými hodnotami, z nichž dvě jsou nastavitelné a jedna pevně stanovena na 100 %. WTD mechanismus zahazuje určité pakety podle asociace k jednotlivým prahům ještě dříve, než se zaplní fronta. [2], [3]

3.2.6 Příkazy pro konfiguraci MLS QoS

1) Zapnutí QoS na přepínači

- (global) **mls qos**

2) Nastavení důvěry

- (interface) **mls qos trust** { **dscp** | **cos** | **ip-precedence** }
  - *dscp* | *cos* | *ip-precedence* – důvěřujeme-li připojenému zařízení, tímto příkazem nastavíme důvěru zvolené hodnotě
- (interface) **mls qos trust device cisco-phone**
  - pokud je k přepínači připojen Cisco IP telefon, můžeme nastavit, aby hodnotě zvolené v předchozím příkazu bylo

důvěřováno pouze tehdy, když je připojen Cisco IP telefon (vyžaduje aktivní CDP protokol)

- (interface) **switchport priority extend cos** *hodnota-cos*
- (interface) **switchport priority extend trust**
  - pokud je ještě k IP telefonem dále připojeno PC, je dobré v telefonu nastavit, aby pakety z PC značkoval určitou hodnotou CoS, kterou poté nastavíme ve směrovači jako důvěryhodnou
  - *hodnota-cos* – námi zvolená hodnota CoS pro pakety z PC za IP telefonem (doporučená hodnota je 0)

3) Nastavení CoS

- (interface) **mls qos cos** *hodnota-cos*
  - *hodnota-cos* – zadáme hodnotu, která má být přidělena neotagovanému příchozímu paketu (defaultně je nastavená hodnota 0)
- (interface) **mls qos cos override**
  - *override* – tento parametr nastaví, aby se CoS hodnoty všech příchozích paketů přepisovaly na hodnotu zvolenou v předchozím příkazu

4) Nastavení mapovacích tabulek

- (global) **mls qos map cos-dscp** *DSCP-hodnoty*
  - *DSCP-hodnoty* – zadáme 8 DSCP hodnot, které budou odpovídat CoS hodnotám 0 – 7.
- (global) **mls qod map dscp-cos** *DSCP-hodnoty* **to** *CoS-hodnota*
  - *DSCP-hodnoty* – zadáme maximálně 8 DSCP hodnot, které následně přiřadíme k jedné CoS hodnotě
  - *CoS-hodnota* – zadáme jednu CoS hodnotu, ke které se přiřadí zadané DSCP hodnoty

5) Nastavení SRR

a) Přiřazení provozu k frontám a prahům

- (global) **mls qos srr-queue** { **input** | **output** } { **cos-map** | **dscp-map** } **queue** *číslo-fronty* **threshold** *číslo-prahu* { *cos-hodnoty* | *dscp-hodnoty* }
  - *input* | *output* – zvolíme, zda půjde o vstupní či výstupní frontu

- *cos-map* | *dscp-map* – zvolíme, podle kterých hodnot se bude provoz třídit
- *číslo-fronty* – zvolíme frontu
- *číslo-prahu* – zvolíme práh
- *cos-hodnoty* | *dscp-hodnoty* – zvolíme konkrétní hodnoty, kterých se bude přiřazování jednat

b) Nastavení prahových hodnot

- (global) **mls qos srr-queue input threshold** *číslo-vstupní-fronty nastavení-prahu-1 nastavení-prahu-2*
  - *nastavení-prahu* – nastavíme procentuální hodnotu pro prahy 1 a 2 (práh 3 bývá nastaven na 100 %)
- (global) **mls qos queue-set output** *číslo-setu* **threshold** *číslo-výstupní-fronty nastavení-prahu-1 nastavení-prahu-2 hodnota-rezervace-pro-frontu maximální-hodnota-fronty*

c) Nastavení velikosti front

- (global) **mls qos srr-queue input buffers** *velikost-fronty-1 velikost-fronty-2*
  - *velikost-fronty* – nastavíme procentuální hodnoty velikostí jednotlivých vstupních front (defaultní nastavení je fronta 1 = 90 %, fronta 2 = 10 %)
- (global) **mls qos queue-set output** *číslo-setu* **buffers** *velikost-fronty-1 velikost-fronty-2 velikost-fronty-3 velikost-fronty-4*
  - *velikost-fronty* – nastavíme procentuální hodnoty velikostí jednotlivých výstupních front (defaultní nastavení je 25 % pro každou frontu)

d) Nastavení váhy pro vstupní fronty

- (global) **mls qos srr-queue input bandwidth** *hodnota-pro-frontu-1 hodnota-pro-frontu-2*
  - *hodnota-pro-frontu* – váhy pro vstupní fronty se nastavují globálně (defaultní nastavení fronta 1 = ½ pásma, fronta 2 = ½ pásma)
- (global) **mls qos srr-queue input priority-queue** *číslo-fronty* **bandwidth** *vyhrazená-šířka-pásma*

- *číslo-fronty* – fronta, kterou chceme zvolit jako prioritní, která je upřednostňována při zahlcení (defaultně je jako prioritní nastavena vstupní fronta 2)
- *vyhrazená-šířka-pásma* – procentuální vyjádření vyhrazené šířky pásma pro prioritní frontu (maximální hodnota je 40 %)

e) Nastavení váhy pro výstupní fronty

- (interface) **queue-set** *číslo-setu*
  - *číslo-setu* – aplikujeme na jednotlivá rozhraní vytvořené sety (defaultně je ke každému rozhraní přiřazen set 1)
- (interface) **srr-queue bandwidth** { **share** | **shape** } *hodnota-pro-frontu-1 hodnota-pro-frontu-2 hodnota-pro-frontu-3 hodnota-pro-frontu-4*
  - *shape* | *share* – zvolíme SRR mód pro dané rozhraní
  - *hodnota-pro-frontu* – vyhrazená šířka pásma pro každou frontu
- (interface) **priority-queue out**
  - tímto příkazem zvolíme frontu 1 jako prioritní, čímž ale způsobíme, že veškeré dříve nastavené hodnoty pro tuto frontu budou ignorovány

6) Další užitečné příkazy

a) Omezení rychlosti rozhraní

- (interface) **srr-queue bandwidth limit** *procent*
  - *procent* – procentuální nastavení rychlosti v rozsahu 10 – 90 %

b) Ověření stavu QoS

- (global) **show mls qos**

c) Kontrola konfigurace QoS na daném rozhraní

- (global) **show mls qos interface** *název-rozhraní*

d) Kontrola konfigurace setů front

- (global) **show mls qos queue-set**

e) Kontrola mapy přiřazení DSCP a CoS hodnot k frontám

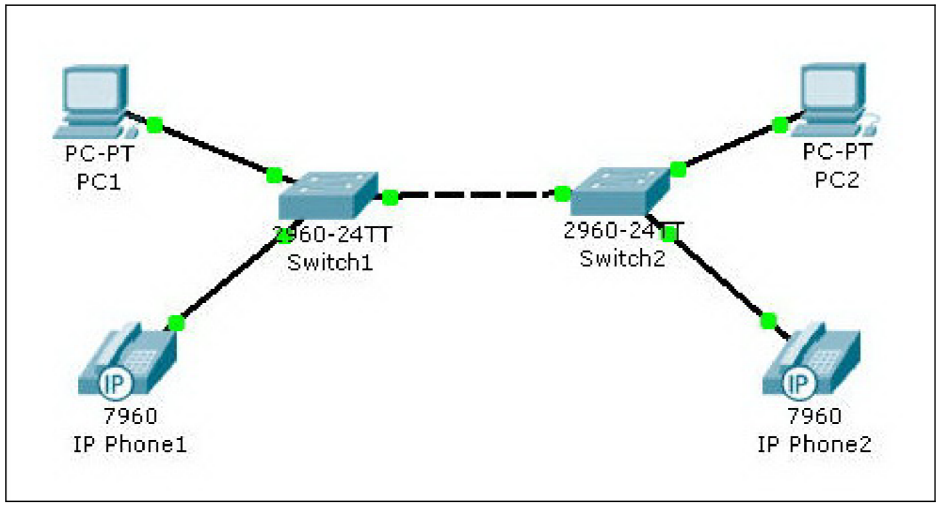
- (global) **show mls qos maps** [**dscp-output-q** | **cos-output-q**]

4 Příklady konfigurace QoS

V této kapitole uvedu přesné konfigurační skripty odpovídající výše uvedeným typům implementace QoS.

4.1 Konfigurace MLS QoS

Tato ukázková konfigurace byla použita v přepínači umístěném v síti podle obrázku 4.1. Před samotnou konfigurací QoS nejdříve nastavuji základní zabezpečovací hesla pro přístup do konfiguračního terminálu přepínače. Celkový postup je znázorněn na obrázku 4.2.



Obr. 4.1: Síť pro konfiguraci MLS QoS

```
Would you like to terminate autoinstall? [yes]: n
Switch>
00:01:21: %LINK-5-CHANGED: Interface Vlan1, changed state to administratively do
wn
Switch>enable
Switch#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#
Switch(config)#hostname Switch1
Switch1(config)#
Switch1(config)#line vty 0 4
Switch1(config-line)#pass switch1
Switch1(config-line)#login
Switch1(config-line)#exit
Switch1(config)#
Switch1(config)#line con 0
Switch1(config-line)#pass switch1
Switch1(config-line)#login
Switch1(config-line)#exit
Switch1(config)#
Switch1(config)#enable secret cisco
Switch1(config)#
```

Obr. 4.2: Konfigurace hesel

Následně je nastavuji VLANy pro jednotlivé datové toky. Nastavení všech VLANů je vidět na obrázku 4.3. VLAN 10 slouží pro správu přepínače, VLAN 20 je pro datový tok IP telefonu a VLAN 30 pro PC. Obrázek 4.4 ukazuje jednak aktivaci samotné kvality služeb na přepínači a dále konfiguraci mapy třídy a mapy politiky s nastavením dscp hodnoty pro datový tok z PC na hodnotu AF11. Po základní konfiguraci QoS dále nastavuji mechanismus front. U použitého přepínače se jedná o mechanismus SRR, celá konfigurace je na obrázku 4.5. Na prvním řádku je nastavení CoS-DSCP mapy, následuje přiřazení CoS a DSCP hodnot k jednotlivým frontám a jejich prahům a nakonec nastavení queue-setu. V posledním kroku nastavuji jednotlivá rozhraní přepínače, viz obrázek 4.6. Každé rozhraní se musí přepnout do trunk módu

```
Switch1(config)#vlan 10
Switch1(config-vlan)#name sprava
Switch1(config-vlan)#exit
Switch1(config)#
Switch1(config)#vlan 20
Switch1(config-vlan)#name voip
Switch1(config-vlan)#exit
Switch1(config)#
Switch1(config)#vlan 30
Switch1(config-vlan)#name provoz_internet
Switch1(config-vlan)#exit
Switch1(config)#
Switch1(config)#interface vlan 10
Switch1(config-if)#ip address 192.168.1.10 255.255.255.0
Switch1(config-if)#no shutdown
Switch1(config-if)#exit
Switch1(config)#
00:02:45: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan10, changed state
to down
```

Obr. 4.3: Konfigurace VLAN

```
Switch1(config)#mls qos
Switch1(config)#
Switch1(config)#access-list 10 permit any
Switch1(config)#
Switch1(config)#class-map match-all trida_internet
Switch1(config-cmap)#match access-group 10
Switch1(config-cmap)#exit
Switch1(config)#
Switch1(config)#policy-map policy_internet
Switch1(config-pmap)#class trida_internet
Switch1(config-pmap-c)#set dscp af11
Switch1(config-pmap-c)#exit
Switch1(config-pmap)#exit
Switch1(config)#
```

Obr. 4.4: Konfigurace map

nebo přístupového módu. Poté zaktivuji příslušnou VLAN, aplikuji politiku a nastavím módy a prioritní frontu SRR mechanismu. Na rozhraní fast ethernet 0/2, ke kterému je připojen Cisco IP telefon, musím nastavit voice mód VLANu

```
Switch1(config)#mls qos map cos-dscp 0 8 16 26 32 46 48 56
Switch1(config)#mls qos srr-queue output cos-map queue 1 threshold 3 5
Switch1(config)#mls qos srr-queue output cos-map queue 2 threshold 1 2 4
Switch1(config)#mls qos srr-queue output cos-map queue 2 threshold 2 3
Switch1(config)#mls qos srr-queue output cos-map queue 2 threshold 3 6 7
Switch1(config)#mls qos srr-queue output cos-map queue 3 threshold 3 0
Switch1(config)#mls qos srr-queue output cos-map queue 4 threshold 3 1
Switch1(config)#mls qos srr-queue output dscp-map queue 1 threshold 1 34
Switch1(config)#mls qos srr-queue output dscp-map queue 1 threshold 3 46
Switch1(config)#$put dscp-map queue 2 threshold 1 16 18 20 22 25 32 36 3
Switch1(config)#mls qos srr-queue output dscp-map queue 2 threshold 2 24 26
Switch1(config)#mls qos srr-queue output dscp-map queue 2 threshold 3 48 56
Switch1(config)#mls qos srr-queue output dscp-map queue 3 threshold 3 0
Switch1(config)#mls qos srr-queue output dscp-map queue 4 threshold 1 8
Switch1(config)#$-queue output dscp-map queue 4 threshold 3 10 12 14
Switch1(config)#mls qos queue-set output 1 threshold 1 70 100 100 100
Switch1(config)#mls qos queue-set output 1 threshold 2 70 80 100 100
Switch1(config)#mls qos queue-set output 1 threshold 4 40 100 100 100
```

Obr. 4.5: Konfigurace front

```
Switch1(config)#interface fa0/1
Switch1(config-if)#description PC
Switch1(config-if)#switchport mode access
Switch1(config-if)#switchport access vlan 30
Switch1(config-if)#service-policy input policy_internet
Switch1(config-if)#srr-queue bandwidth share 1 70 25 5
Switch1(config-if)#srr-queue bandwidth shape 30 0 0 0
Switch1(config-if)#priority-queue out
Switch1(config-if)#exit
Switch1(config)#interface fa0/2
Switch1(config-if)#description VoIP
Switch1(config-if)#switchport mode access
Switch1(config-if)#switchport access vlan 20
Switch1(config-if)#switchport voice vlan 20
Switch1(config-if)#srr-queue bandwidth share 1 70 25 5
Switch1(config-if)#srr-queue bandwidth shape 30 0 0 0
Switch1(config-if)#priority-queue out
Switch1(config-if)#mls qos trust cos
Switch1(config-if)#mls qos trust device cisco-phone
Switch1(config-if)#exit
Switch1(config)#interface fa0/3
Switch1(config-if)#description Switch2
Switch1(config-if)#switchport mode trunk
Switch1(config-if)#switchport trunk allowed vlan 1-31
Switch1(config-if)#switchport nonegotiate
Switch1(config-if)#srr-queue bandwidth share 1 70 25 5
Switch1(config-if)#srr-queue bandwidth shape 30 0 0 0
Switch1(config-if)#priority-queue out
Switch1(config-if)#mls qos trust dscp
Switch1(config-if)#exit
```

Obr. 4.6: Konfigurace portů

20 a důvěru k CoS hodnotám z Cisco IP telefonu. Při tomto nastavení je nutné, aby byl aktivní CDP protokol.

Po konfiguraci všech náležitých mechanismů se doporučuje nastavení zkontrolovat pomocí *show* příkazů. Na obrázku 4.7 je vidět výpis příkazu *show mls qos*, který slouží k ověření stavu QoS na daném zařízení. Na obrázcích 4.8 – 4.10 je výpis příkazu *show mls qos maps*. Tento příkaz zobrazí kompletní nastavení DSCP a CoS hodnot v jednotlivých mapách. Poslední příkaz, jehož výpis zde uvedu, je na obrázku 4.11, jedná se o příkaz *show mls qos queue-set* a slouží pro kontrolu nastavení jednotlivých queue-setů. [1], [2]

```
Switch1#show mls qos
QoS is enabled
QoS ip packet dscp rewrite is enabled
```

Obr. 4.7: Show mls qos

```
Policed-dscp map:
  d1 :  d2 0  1  2  3  4  5  6  7  8  9
  ---------------------------------------
   0 :    00 01 02 03 04 05 06 07 08 09
   1 :    10 11 12 13 14 15 16 17 18 19
   2 :    20 21 22 23 24 25 26 27 28 29
   3 :    30 31 32 33 34 35 36 37 38 39
   4 :    40 41 42 43 44 45 46 47 48 49
   5 :    50 51 52 53 54 55 56 57 58 59
   6 :    60 61 62 63

Dscp-cos map:
  d1 :  d2 0  1  2  3  4  5  6  7  8  9
  ---------------------------------------
   0 :    00 00 00 00 00 00 00 00 01 01
   1 :    01 01 01 01 01 01 02 02 02 02
   2 :    02 02 02 02 03 03 03 03 03 03
   3 :    03 03 04 04 04 04 04 04 04 04
   4 :    05 05 05 05 05 05 05 05 06 06
   5 :    06 06 06 06 06 06 07 07 07 07
   6 :    07 07 07 07
```

Obr. 4.8: Show mls qos maps

```
Cos-dscp map:
     cos:   0  1  2  3  4  5  6  7
  --------------------------------
    dscp:   0  8 16 26 32 46 48 56

IpPrecedence-dscp map:
  ipprec:   0  1  2  3  4  5  6  7
  --------------------------------
    dscp:   0  8 16 24 32 40 48 56

Dscp-outputq-threshold map:
  d1 :d2    0     1     2     3     4     5     6     7     8     9
  ---------------------------------------------------------------
   0 :    03-03 02-01 02-01 02-01 02-01 02-01 02-01 02-01 04-01 02-01
   1 :    04-03 02-01 04-03 02-01 04-03 02-01 02-01 03-01 02-01 03-01
   2 :    02-01 03-01 02-01 03-01 02-02 02-01 02-02 03-01 03-01 03-01
   3 :    03-01 03-01 02-01 04-01 01-01 04-01 02-01 04-01 04-01 04-01
   4 :    01-01 01-01 01-01 01-01 01-01 01-01 01-03 01-01 02-03 04-01
   5 :    04-01 04-01 04-01 04-01 04-01 04-01 02-03 04-01 04-01 04-01
   6 :    04-01 04-01 04-01 04-01

Dscp-inputq-threshold map:
  d1 :d2    0     1     2     3     4     5     6     7     8     9
   0 :    01-01 01-01 01-01 01-01 01-01 01-01 01-01 01-01 01-01 01-01
   1 :    01-01 01-01 01-01 01-01 01-01 01-01 01-01 01-01 01-01 01-01
   2 :    01-01 01-01 01-01 01-01 01-01 01-01 01-01 01-01 01-01 01-01
   3 :    01-01 01-01 01-01 01-01 01-01 01-01 01-01 01-01 01-01 01-01
   4 :    02-01 02-01 02-01 02-01 02-01 02-01 02-01 02-01 01-01 01-01
   5 :    01-01 01-01 01-01 01-01 01-01 01-01 01-01 01-01 01-01 01-01
   6 :    01-01 01-01 01-01 01-01
```

Obr. 4.9: Show mls qos maps

```
 Cos-outputq-threshold map:
            cos:  0   1   2   3   4   5   6   7
            ------------------------------------
queue-threshold: 3-3 4-3 2-1 2-2 2-1 1-3 2-3 2-3

 Cos-inputq-threshold map:
            cos:  0   1   2   3   4   5   6   7
            ------------------------------------
queue-threshold: 1-1 1-1 1-1 1-1 1-1 2-1 1-1 1-1


 Dscp-dscp mutation map:
 Default DSCP Mutation Map:
   d1 :  d2 0  1  2  3  4  5  6  7  8  9
   ---------------------------------------
    0 :    00 01 02 03 04 05 06 07 08 09
    1 :    10 11 12 13 14 15 16 17 18 19
    2 :    20 21 22 23 24 25 26 27 28 29
    3 :    30 31 32 33 34 35 36 37 38 39
    4 :    40 41 42 43 44 45 46 47 48 49
    5 :    50 51 52 53 54 55 56 57 58 59
    6 :    60 61 62 63
```

Obr. 4.10: Show mls qos maps

```
Switch1#show mls qos queue-set
Queueset: 1
Queue     :       1       2       3       4
----------------------------------------------
buffers   :      25      25      25      25
threshold1:      70      70     100      40
threshold2:     100      80     100     100
reserved  :     100     100      50     100
maximum   :     100     100     400     100
Queueset: 2
Queue     :       1       2       3       4
----------------------------------------------
buffers   :      25      25      25      25
threshold1:     100     200     100     100
threshold2:     100     200     100     100
reserved  :      50      50      50      50
maximum   :     400     400     400     400
```

Obr. 4.11: Show mls qos queue-set

4.2 Konfigurace AutoQoS

Nyní již popíši pouze samotnou konfiguraci QoS a s ní spojených mechanismů. Konfigurace probíhá ve stejné síti jako v předešlém případě (viz obrázek 4.1). AutoQoS stačí nakonfigurovat pouze na rozhraní, ke kterému je připojen Cisto IP telefon, přičemž veškerá manuální konfigurace spočívá v aktivaci QoS. O zbytek konfigurace se stará samo zařízení.

Celková konfigurace i s automaticky vygenerovanými příkazy [3]:

- (config-if) auto qos voip cisco-phone
- (config) mls qos
- (config) mls qos map cos-dscp 0 8 16 26 32 46 48 56
- (config) no mls qos srr-queue input cos-map
- (config) mls qos srr-queue input cos-map queue 1 threshold 3 0
- (config) mls qos srr-queue input cos-map queue 1 threshold 2 1
- (config) mls qos srr-queue input cos-map queue 2 threshold 1 2
- (config) mls qos srr-queue input cos-map queue 2 threshold 2 4 6 7
- (config) mls qos srr-queue input cos-map queue 2 threshold 3 3 5
- (config) mls qos map cos-dscp 0 8 16 26 32 46 48 56
- (config) no mls qos srr-queue output cos-map
- (config) mls qos srr-queue ouput cos-map queue 1 threshold 3 5
- (config) mls qos srr-queue ouput cos-map queue 2 threshold 3 3 6 7
- (config) mls qos srr-queue output cos-map queue 3 threshold 3 2 4
- (config) mls qos srr-queue output cos-map queue 4 threshold 2 1
- (config) mls qos srr-queue output cos-map queue 4 threshold 3 0
- (config) no mls qos srr-queue input dscp-map
- (config) mls qos srr-queue input dscp-map queue 1 threshold 2 9 10 11 12 13 14 15
- (config) mls qos srr-queue input dscp-map queue 1 threshold 3 0 1 2 3 4 5 6 7
- (config) mls qos srr-queue input dscp-map queue 1 threshold 3 32
- (config) mls qos srr-queue input dscp-map queue 2 threshold 1 16 17 18 19 20 21 22 23

- (config) mls qos srr-queue input dscp-map queue 2 threshold 2 33 34 35 36 37 38 39 48
- (config) mls qos srr-queue input dscp-map queue 2 threshold 2 49 50 51 52 53 54 55 56
- (config) mls qos srr-queue input dscp-map queue 2 threshold 2 57 58 59 60 61 62 63
- (config) mls qos srr-queue input dscp-map queue 2 threshold 3 24 25 26 27 28 29 30 31
- (config) mls qos srr-queue input dscp-map queue 2 threshold 3 40 41 42 43 44 45 46 47
- (config) no mls qos srr-queue output dscp-map
- (config) mls qos srr-queue output dscp-map queue 1 threshold 3 40 41 42 43 44 45 46 47
- (config) mls qos srr-queue output dscp-map queue 2 threshold 3 24 25 26 27 28 29 30 31
- (config) mls qos srr-queue output dscp-map queue 2 threshold 3 48 49 50 51 52 53 54 55
- (config) mls qos srr-queue output dscp-map queue 2threshold 3 56 57 58 59 60 61 62 63
- (config) mls qos srr-queue output dscp-map queue 3 threshold 3 16 17 18 19 20 21 22 23
- (config) mls qos srr-queue output dscp-map queue 3 threshold 3 32 33 34 35 36 37 38 39
- (config) mls qos srr-queue output dscp-map queue 4 threshold 1 8
- (config) mls qos srr-queue output dscp-map queue 4 threshold 2 9 10 11 12 13 14 15
- (config) mls qos srr-queue output dscp-map queue 4 threshold 3 0 1 2 3 4 5 6 7
- (config) no mls qos srr-queue input priority-queue 1
- (config) no mls qos srr-queue input priority-queue 2
- (config) mls qos srr-queue input bandwidth 90 10
- (config) mls qos srr-queue input threshold 1 8 16
- (config) mls qos srr-queue input threshold 2 34 66
- (config) mls qos srr-queue input buffers 67
- (config) mls qos queue-set output 1 threshold 1 138 138 92 138
- (config) mls qos queue-set output 1 threshold 2 138 138 92 400
- (config) mls qos queue-set output 1 threshold 3 36 77 100 318
- (config) mls qos queue-set output 1 threshold 4 20 50 67 400
- (config) mls qos queue-set output 2 threshold 1 149 149 100 149
- (config) mls qos queue-set output 2 threshold 2 118 118 100 235
- (config) mls qos queue-set output 2 threshold 3 41 68 100 272
- (config) mls qos queue-set output 2 threshold 4 42 72 100 242
- (config) mls qos queue-set output 1 buffers 10 10 26 54
- (config) mls qos queue-set output 1 buffers 16 6 17 61
- (config-if) priority-queue out
- (config-if) srr-queue bandwidth share 10 10 60 20
- (config-if) mls qos trust cos
- (config-if) mls qos trust dscp
- (config-if) mls qos trust device cisco-phone
- (config) mls qos map policed-dscp 24 26 46 to 0
- (config) class-map match-all AutoQoS-VoIP-RTP-Trust
- (config-cmap) match ip dscp ef
- (config) class-map match-all AutoQoS-VoIP-Control-Trust
- (config-cmap) match ip dscp cs3 af31
- (config) policy-map AutoQoS-Police-CiscoPhone
- (config-pmap) class AutoQoS-VoIP-RTP-Trust
- (config-pmap-c) set dscp ef
- (config-pmap-c) police 320000 8000 exceed-action policed-dscp-transmit
- (config-pmap) class AutoQoS-VoIP-Control-Trust
- (config-pmap-c) set dscp cs3
- (config-pmap-c) police 32000 8000 exceed-action policed-dscp-transmit
- (config-if) service-policy input AutoQoS-Police-CiscoPhone

4.3 Konfigurace RSVP

Tuto ukázkovou konfiguraci provedu na sériovém rozhraní směrovače. Nejprve nastavím mechanismus LLQ a mechanismus správy zahlcení WFQ. Dále na rozhraní nastavím maximální velikost rezervace 512 kbps, maximální rezervaci pro jeden datový tok 30 kbps a uhlazení špičky na 400 %, což znamená minimální vyhlazování. Příkazem *ip rsvp reservation* nastavím ukázkovou proxy rezervaci z adresy 10.1.1.10 na 10.2.2.20 s dalším krokem 10.1.1.1, který se nachází za sériovým rozhraním 0/0/0 . Jde o druh rezervace *fixed filter* se zaručenou rychlostí 20 kbps a špičkou 80 kB. Nakonec ještě nastavím dva COPS severy a příkaz pro zasílání veškerých rozhodnutí protokolu RSVP. [5] [6]

- (config) ip rsvp pq-profile voice-like
- (config) interface serial 1/0/0
- (config-if) fair-queue
- (config-if) ip rsvp bandwidth 512 30
- (config-if) burst policing 400
- (config) ip rsvp reservation 10.1.1.10 10.2.2.20 udp 40 50 10.1.1.1 seriál 0/0/0 ff rate 20 80
- (config) ip rsvp policy cops servers 192.168.10.10 192.168.10.11
- (config) ip rsvp policy cops report-all

5 Závěr

Jak jsem již zmínil v úvodu, původně jsem měl tuto práci vypracovat ve spolupráci s firmou Maxprogres. V průběhu zpracování práce zájem firmy opadl, což vypracování zkomplikovalo. Neměl jsem k dispozici žádné reálné síťové prvky a po dohodě s vedoucím práce jsem chtěl veškeré simulace provádět v programu Packet Tracer od firmy Cisco. Tento program ale, bohužel, žádné konfigurační příkazy pro QoS nepodporuje. Nakonec jsem získal přístup ke směrovačům řady Catalyst 2960, kde jsem si mohl alespoň vyzkoušet konfiguraci MLS QoS. Tyto směrovače pracují pouze s vrstvou L2 a navíc jsem neměl k dispozici IP telefony, tudíž jsem vyzkoušel pouze samotnou konfiguraci. Zmiňovaná praktická konfigurace je popsána v kapitole 4.1. Velikým přínosem pro mě byla možnost zorientovat se v celkové problematice kvality služeb a jejích nejnovějších konfiguračních trendů. Pokud bych měl zhodnotit jednotlivé konfigurace, u sítí zaměřených na služby VoIP bych upřednostnilvyužití konfigurace AutoQoS, kde si zařízení samo optimálně nastaví veškeré parametry. Pokud bychom chtěli k VoIP ještě přidat například IPTV, bude vhodnější konfigurace manuální.

Použitá literatura:

[1] ADÁMEK, David. Implementace QoS v přístupové síti, 2008. 60 s. Diplomová práce

[2] BOUŠKA, Petr. QoS - Quality of Service. Samuraj-cz [online]. 2009 [cit. 2009-05-19]. Dostupný z WWW: <http://www.samuraj-cz.com/serie/qos-quality-of-service/>.

[3] Catalyst 2960 Switch Software Configuration Guide : Configuring QoS [online]. - [cit. 2009-05-29]. Dostupný z WWW: <http://cisco.biz/en/US/docs/switches/lan/catalyst2960/software/release/12.2_40_se/configuration/guide/swqos.html#wpxref86220>.

[4] ČÍKA, Petr. Multimediální služby, 2007. 106 s.

[5] DOVICA, Marek, KONÁR, Milan. Protokol RSVP na Cisco Routerech : Semestrální projekt. Projekty [online]. - [cit. 2009-05-15], s. 1-11. Dostupný z WWW: <http://www.cs.vsb.cz/grygarek/SPS/projekty0506/RSVP-Cisco.pdf>.

[6] HUCABY, David, MCQUERRY, Steve. Konfigurace směrovačů Cisco : Autorizovaný výukový průvodce. 1. vyd. Libor Pácl, Jiří Veselský. Brno : Computer Press, 2004. ISBN 80-722-6951-8. Quality of Service, s. 347-390.

[7] KACÁLEK, Jan. Modely pro zajištění kvality služeb v IP sítích [online]. 2006 [cit. 2008-12-01]. Dostupný z WWW: <http://amarok.cesketelekomunikace.cz/xkacal00/index.php?action=intro>.

[8] QoS v datových sítích IntServ a DiffServ. Přednášky ZČU [online]. 2007 [cit. 2008-12-12]. Dostupný z WWW: <http://www.kiv.zcu.cz/~ledvina/Prednasky-PSI-2007/qos-text.pdf>.

[9] RFC791 - Internet Protocol [online]. 1981 [cit. 2008-11-24]. Dostupný z WWW: <http://www.faqs.org/rfcs/rfc791.html>.

[10] RFC1349 - Type of Service in the Internet Protocol Suite [online]. 1992 [cit. 2008-12-01]. Dostupný z WWW: <http://rfc.sunsite.dk/rfc/rfc1349.html>.

[11] RFC1633 - Integrated Services in the Internet Architecture [online]. 1994 [cit. 2008-11-01]. Dostupný z WWW: <http://www.ifla.org.sg/documents/rfcs/rfc1633.txt>.

[12] RFC2205 - Resource ReSerVation Protocol (RSVP) [online]. 1997 [cit. 2008-11-09]. Dostupný z WWW: <http://www.faqs.org/rfcs/rfc2205.html>.

[13] RFC2460 - Internet Protocol, Version 6 (IPv6) Specification [online]. 1998 [cit. 2008-12-01]. Dostupný z WWW: <http://www.faqs.org/rfcs/rfc2460.html>.

[14] RFC2474 - Definition of the Differentiated Services Field (DS Field) [online]. 1998 [cit. 2008-11-20]. Dostupný z WWW: <http://www.ietf.org/rfc/rfc2474.txt>.

[15] RFC2475 - An Architecture for Differentiated Service [online]. 1998 [cit. 2008-11-25]. Dostupný z WWW: <http://www.faqs.org/rfcs/rfc2475.html>.

[16] RFC2836 - Per Hop Behavior Identification Codes [online]. 2000 [cit. 2008-12-02]. Dostupný z WWW: <http://www.faqs.org/rfcs/rfc2836.html>.

[17] RFC3260 - New Terminology and Clarifications for Diffserv [online]. 2002 [cit. 2008-11-10]. Dostupný z WWW: <http://www.faqs.org/rfcs/rfc3260.html>.

[18] SZIGETI, Tim, HATTINGH, Christina. End-to-End QoS Network Design : Quality of Service in LANs, WANs and VPNs : Cisco Press, 2004. 734 s.

[19] Wikipedia - Quality of Services [online]. 2008 [cit. 2008-11-10]. Dostupný z WWW: <http://en.wikipedia.org/wiki/QoS>.

Seznam zkratek

ACL – Access List

AF – Assured Forwarding

CBWFQ – Class-Based Weighted Fair Queue

CDP – Cisco Discovery Protocol

CLP – Cell loss Priority

CoS – Class of Services

COPS – Common Open Policy Service

DE – Discard Eligibility

DiffServ – Differentiated Service

DSCP – DiffServ Code Point

EF – Expedited Forwarding

FF – Fixed Filter

GTS – Generic Traffic Shaping

IntServ – Integrated Service

IP – Internet Protocol

IPP – IP Precedence

LLQ – Low Latency Queueing

MLS – Multilayer Switch

MQC – Modular QoS Command Line Interface

PDP – Policy Decision Points

PEP – Policy Enforcement Points

PHB – Per Hop Behaviour

QoS – Quality of Service

RTI – Real Time Intolerant

RTP – Real Time Protocol

RTT – Real Time Tolerant

RSVP – Reservation Protocol

SE – Shared Explicit

SLA – Service Level Agreements

SRR – Shaped/Shared Round Robin

TC – Traffic Class

TCA – Traffic Confitioning Agreements

TCP – Transmission Control Protocol

ToS – Type of Service

TTL – Time to Live

UDP – User Datagram Protocol

VoD – Video on Demand

VoIP – Voice over IP

WF – Wildcard filter

WFQ – Weighted Fair Queue

WRED – Weighted Random Early Detection

WRR – Weighted Round Robin

WTD – Weighted Tail Drop