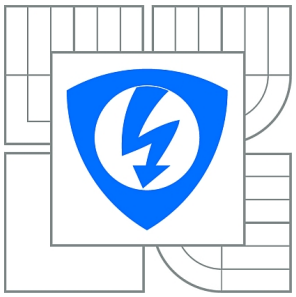
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

DEPARTMENT OF TELECOMMUNICATIONS

ZAJIŠTĚNÍ KVALITY SLUŽEB V IP SÍTÍCH

QUALITY OF SERVICE ASSURANCE IN IP NETWORKS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

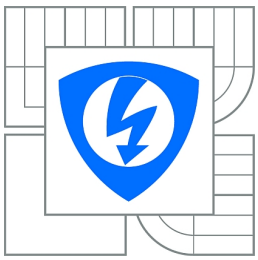
AUTOR PRÁCE — LUKÁŠ KUPČIHA

AUTHOR

VEDOUCÍ PRÁCE — Ing. JIŘÍ HOŠEK

SUPERVISOR

BRNO 2010



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor
Teleinformatika

Student:	Lukáš Kupčiha	***ID:***	20323
Ročník:	3	***Akademický rok:***	2009/2010

NÁZEV TÉMATU:

Zajištění kvality služeb v IP sítích

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s možnostmi konfigurace mechanizmů pro zajištění kvality služeb (QoS) na síťových prvcích firmy Cisco. Vytvořte laboratorní síť tvořenou Cisco prvky. V této síti proveďte konfiguraci mechanizmu DiffServ sloužícího pro zajištění kvality služeb v IP sítích. V takto nakonfigurované síti generujte provoz a následně analyzujte vliv použitého QoS mechanizmu na rozložení síťového provozu a jeho základní charakteristiky. Dále ve vytvořené síti realizujte několik konfiguračních scénářů, pomocí kterých se pokuste nalézt optimální konfiguraci mechanizmu DiffServ. Veškeré dosažené výsledky přehledně zdokumentujte v podobě závěrečné práce.

DOPORUČENÁ LITERATURA:

[1] DOOLEY, K., BROWN, I.: Cisco IOS Cookbook. Sebastopol: O'Reilly Media, 2006, ISBN: 978-0596527228.
[2] SZIGETI, T., HATTINGH, C.: End-to-end QoS network design. Indianapolis: Cisco Press, 2005, ISBN: 1-58705-176-1.

Termín zadání:	29.1.2010	***Termín odevzdání:***	2.6.2010

Vedoucí práce: Ing. Jiří Hošek

prof. Ing. Kamil Vrba, CSc.
Předseda oborové rady

UPOZORNĚNÍ:

ANOTACE

Cílem této práce bylo seznámit se s možnostmi konfigurace mechanismů pro zajištění kvality služeb (QoS) na síťových prvcích firmy Cisco. Byla prozkoumána a vysvětlena problematika jednotlivých technik pro kontrolu a řízení přenosové šířky pásma, zpoždění, kolísání zpoždění a ztrátovosti paketů. Pozornost poté byla věnována především mechanismu DiffServ. Detailně jsou probrány jeho jednotlivé oblasti působení: rozlišení a označení provozu, správa datového toku a jeho omezování a vyhrazování, správa zahlcení pomocí front a zabránění zahlcení pomocí řízeného zahazování. Dále byly prozkoumány možnosti implementace QoS na zařízeních Cisco. Pomocí Cisco MQC pak byly demonstrovány základní principy a příkazy pro zajišťování kvality služeb. Vysvětleny a převedeny byly pojmy jako: definování mapy tříd, definování mapy politik a aplikování mapy politik. Následně byly sestaveny scénáře měření a sestavena testovací síť. Proběhlo měření vlivu mechanismů FIFO, PQ, CBWFQ a WRED na síťový provoz. Na konci práce pak byl vybrán nejoptimálnější mechanismus zajištění kvality služeb.

KLÍČOVÁ SLOVA

CBWFQ, Cisco, FIFO, DiffServ, DSCP, MQC, IP, PQ, QoS, RED, Token Bucket, WRED

ANNOTATION

The aim of this work was introducing the options of configuration mechanisms for quality of service (QoS) in basic network products of Cisco company. The characteristics of the techniques for control and management of bandwidth, delay, jitter and packet loss were examined and explained. The attention was then focused mainly on DiffServ mechanism. Its individual functions, such as classification and marking, traffic rate management and its policing and shaping, congestion management queuing and congestion avoidance, are discussed in detail. Further investigation was aimed at the possibility of QoS implementation on Cisco devices. The basic principles and commands for providing the quality of service were then demonstrated by Cisco MQC. The specific terms, such as traffic class definition, policy definition and policy application, were presented and explained. Consequently, the different scenarios of measurement were proposed and a testing network was designed. Influence of FIFO, PQ, CBWFQ and WRED mechanisms on the network traffic was measured. As a result of the work, the optimal mechanism for providing the quality of service was chosen.

KEYWORDS

CBWFQ, Cisco, FIFO, DiffServ, DSCP, MQC, IP, PQ, QoS, RED, Token Bucket, WRED

BIBLIOGRAFICKÁ CITACE PRÁCE

KUPČIHA, L. *Zajištění kvality služeb v IP sítích.* Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2010. 78 s. Vedoucí bakalářské práce Ing. Jiří Hošek.

PROHLÁŠENÍ O PŮVODNOSTI PRÁCE

Prohlašuji, že svou bakalářskou práci na téma „Zajištění kvality služeb v IP sítích“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce 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 30. 5. 2010

podpis autora

PODĚKOVÁNÍ

Děkuji vedoucímu bakalářské práce Ing. Jiřímu Hoškovi za výstižné připomínky, cenné podněty ke zlepšení a zejména trpělivost a ochotu řešit vše, co bylo potřeba.

V Brně dne 30. 5. 2010

podpis autora

OBSAH

ÚVOD

Síťové technologie zažívají v několika posledních letech nevídaný rozmach. Setkat se s nimi můžeme v podnicích a organizacích všech velikostí. Pro mnoho z nich je životně důležité nejen to, aby se všechen datový provoz dostal bezpečně tam, kam má, ale také tehdy kdy má.

Cílem bakalářské práce je prozkoumat možnosti konfigurace mechanizmů pro zajištění kvality služeb. V praktické části práce se pak především zaměřuje na implementaci nástrojů pro řazení do front a předcházení zahlcení na síťových prvcích Cisco.

Co si však vlastně představit pod pojmem kvalita služeb? Podle [21] můžeme volně říci, že kvalita služeb, označovaná zkratkou QoS (Quality Of Service) je soubor běhových procesů, které aktivně vstupují do řízení šířky pásma s cílem poskytovat zaručené úrovně síťových služeb daným aplikacím nebo uživatelům. Implementovaná technologie QoS představuje základní systém pro definici zásad (politiky) poskytování služeb a rozšíření koncové komunikace na obsluhovaná spojení, a to i přes autonomní systémy.

Jednodušeji řečeno je QoS soubor určitých mechanismů, jejichž základním úkolem je upřednostnění určitého síťového provozu před jiným.

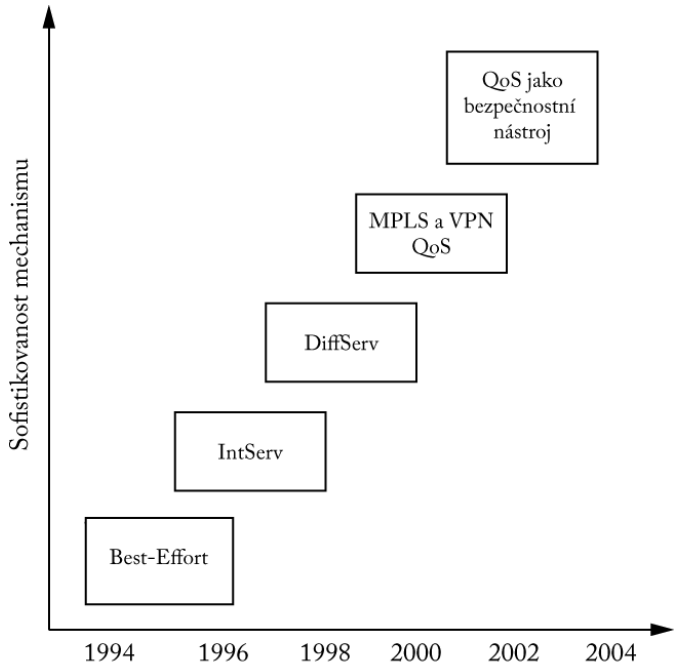
1. VÝVOJ QOS V IP SÍTÍCH

IP (Internet Protocol) sítě v polovině 90. let ve velké většině fungovaly bez jakéhokoliv zajišťování QoS, tzv. Best-Effort mechanismem. Ostatně internet jako celek takto funguje dodnes. Síť se snažila vyhovět všem požadavkům aplikací stejně, neexistoval žádný prioritní provoz. V okamžiku, kdy přestala požadavkům stačit kapacita sítě, začala síť pakety přenášející data jednotlivých aplikací víceméně náhodně zahazovat nebo pozdržovat. To samozřejmě vedlo k aplikačně těžko „ošetřitelným“ výpadkům a zpomalování přenosu [21, 22].

Po tom, co do světa protokolu IP začaly pronikat technologie internetové telefonie, videokonferencí a dalších interaktivních služeb, začala být problematika přidělování kapacity velice sledována. Uživatelé požadovali, aby jim poskytované služby byly na určité kvalitativní úrovni a začali vyžadovat zajištění kvality služeb [21, 22].

Na rozdíl od sítí ATM (Asynchronous Transfer Mode) a Frame-Relay, tedy sítí s danou infrastrukturou, bylo nasazení QoS u IP složitější. V IP síti je paket nucen procházet nehomogenním prostředím s mnoha různými protokoly, čímž se složitost nasazení QoS řádově zvyšuje [22].

Ani tyto problémy ale nezabránily dalším snahám o transformaci, tehdy především privátních sítí, na sofistikovanější model řízení provozu. První krok ke standardizaci QoS provedlo v červnu 1994 sdružení IETF (Internet Engineering Task Force). Vydalo RFC 1633 (Request For Comment) pod názvem IntServ (Integrated Services). Dokument byl zaměřen na signalizační protokol nazývaný RSVP (Resource Reservation Protocol). V srpnu 1998 se k němu připojil nový mechanismus nazývaný DiffServ (Differentiated Services) definovaný v RFC 2475 [2, 20]. Celý proces historického vývoje ukazuje obr. 1.1.



Obr. 1.1: Vývoj mechanismů QoS [20]

1.1 Základní parametry

Základním stavebním kamenem úspěšného zajišťování kvality služeb je určení parametrů, na které jsou různé typy síťového provozu citlivé. Mezi ty nejdůležitější patří zpoždění a kolísání zpoždění, ztrátovost paketů a šířka pásma.

1.1.1 Zpoždění a kolísání zpoždění

Pojmem zpoždění (Delay) označujeme dobu, kterou trvá přenos jinak nepoškozené zprávy od zdroje k příjemci. Samotnou prodlevu ovlivňuje:

- zpoždění způsobené kódováním a přípravou paketů pro přenos,
- rychlost, jakou se šíří signál médiem (přenosové zpoždění),
- doba zpracování směrovači (zpoždění ve frontě na odbavení a další prodleva při nalezení cesty sítí a odpovídajícího výstupního portu) [22].

Celkové zpoždění, dané součtem všech prodlev vyskytujících se během přenosu, označujeme jako End-To-End Delay [21].

Během přenosu ale nemusíme dostávat pakety vždy se stejným zpožděním a ve stejných intervalech. Kolísání zpoždění (Jitter) představuje právě tento časový rozdíl mezi zpožděními jednotlivých paketů. Na rozdíl od zpoždění tedy kolísání představuje narušení integrity přenášené zprávy [17, 22]. U běžně používaných základních aplikací, jako je elektronická pošta nebo webové prohlížeče, není kolísání významným činitelem. Úplně jiná situace ale nastává u hlasových služeb. Uživatelé kolísání vnímají jako nepříjemné výpadky spojení, odmlky atd. Celý problém se pak musí složitě řešit pomocí vyrovnávacích pamětí.

Doba zpoždění i kolísání zpoždění jsou tedy ovlivněny zatížením samotné sítě, volbou přenosového média a směrovače (nebo přepínače). U posledně jmenovaných zařízení je potřeba věnovat pozornost nejen technickým parametrům, ale i celkové koncepci. Pomalé zpracování a dlouhé fronty na odbavení vedou k vyššímu zpoždění, naopak u krátkých může docházet ke ztrátě paketů (jednoduše se do fronty již nevejdou). Řešením jsou směrovače (nebo přepínače) vyšší třídy s architekturou bez vzájemného blokování portů, navíc délka front i velikost vyrovnávací paměti je u těchto zařízení pomocí třídy QoS lehce staticky nastavitelná [17, 21, 22].

1.1.2 Ztrátovost paketů

Pojem ztrátovost paketů lze definovat podílem přijatých a vyslaných paketů, který je následně vyjádřen v procentech. Ke ztrátě paketů může docházet z mnoha příčin. Může se jednat o přeplněné vyrovnávací paměti směrovačů nebo přepínačů, nízkou kvalitu signálu v bezdrátových sítích nebo dokonce o samotné mechanismy QoS záměrně zahazující pakety určitého toku [13, 17].

Problém se ztrátou paketů je zásadní u přenosu klasických dat (souborů). Hraje mnohem důležitější roli než například zpoždění či kolísání, které jsme schopni nějakým způsobem kompenzovat ukládáním paketů do vyrovnávací paměti [22].

1.1.3 Šířka pásma

Z pohledu kvality služeb je šířka pásma přesně daná, neměnitelná. QoS nedokáže její kapacitu navýšit, umožní nám však její hospodárnější využití. Efektivně dokáže využít toho, co má k dispozici, aby maximálně vyhověla požadavkům důležitých datových přenosů [21, 22].

1.2 QoS mechanismy

Dosažení požadované QoS předpokládá spolupráci všech síťových vrstev a koncovou spolupráci síťových prvků přes celou síť. Maximální kvalita služby tedy závisí na nejslabším článku celé přenosové cesty. Na rozdíl od klasických přenosových sítí s přepojování okruhů, které zaručují požadovanou úroveň QoS specifikovanou v SLA (Service Level Agreement), paketové sítě potřebují doplňkové mechanismy. Takovým doplňkem je u Ethernetu značení rámců a jejich řazení do front podle priorit (IEEE 802.1Q/p). Síťový protokol IP je na tom obdobně. Od prvopočátku byl postaven na mechanismu Best-Effort. Jde o klasický TCP (Transmission Control Protocol) provoz, kde je přenosová rychlost a doba doručení proměnlivá a nespecifikovatelná, měnící se podle aktuálního zatížení sítě. Aby tedy mohly být využívány služby QoS v IP sítích musela definovat IETF dva nové mechanismy: IntServ (Integrated Services) a DiffServ (Differentiated Services) [3, 21].

Starším z mechanismů je IntServ. V principu funguje tak, že pro každý individuální datový tok v síti je protokolem RSVP (Resource Reservation Protocol) signalizována požadovaná kvalita služby. Síťová zařízení si na základě signalizace rezervují dostatečné prostředky, aby mohly službu v požadované kvalitě poskytnout. Tím je sice teoreticky možné pro každý takový datový tok zajistit požadovanou kvalitu služby, ale zároveň je nutné, aby u všech zařízení, přes která procházejí IP pakety datových toků (včetně koncových zařízení jako PC a servery), existovala podpora protokolu RVSP. Pokud ještě vezmeme v potaz, že každý směrovač v síti musí udržovat stavovou informaci pro každý datový tok s požadavkem na QoS, bude zřejmé, jak by vypadaly paměťové a výkonnostní nároky na směrovače u rozsáhlejších sítí se statisícovými počty datových toků a jak to vypadá s rozšiřitelností modelu IntServ. Tento

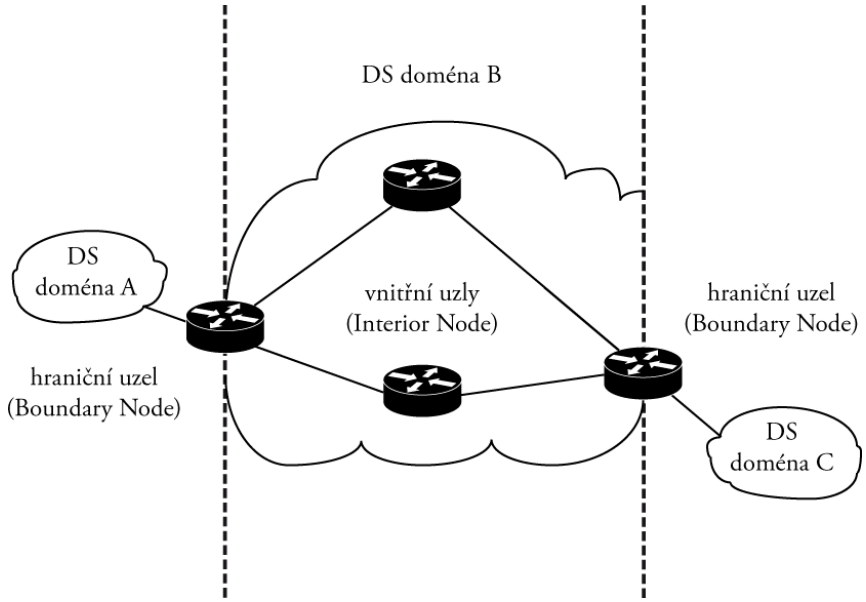
mechanismus tak nalézá uplatnění už jen v menších podnikových sítích pro přenos hlasu pomocí VoIP (Voice over IP), kdy můžeme úspěšně využít před ustavením komunikace kontrolu přijetí spojení (CAC – Connection Admission Control) právě protokolem RSVP. Stále více se tak hlavní metodou pro zajištění QoS stává mechanismus DiffServ, kterému je věnována další část práce [20, 21, 22].

2. MECHANISMUS DIFFSERV

Všechny rozsáhlejší sítě jsou vytvořeny propojováním několika menších sítí. Jde obvykle o velice nehomogenní prostředí, kde každá z těchto menších sítí je většinou odlišně uspořádána, spravována, obsahuje odlišné síťové prvky a tedy i odlišně zpracovává datový provoz. Aby implementace QoS nebyla neřešitelně komplikovaná, je vhodné rozdělit takové sítě na oblasti s autonomní správou, zajišťující alespoň základní společné parametry. V takovéto oblasti pak můžeme snadněji zajistit nasazení QoS, ale především si tak vyčleníme území s takzvanou hranicí důvěry (Trust Boundary). Takovéto území pak nazýváme DiffServ doménou, příkladem může být podniková síť nebo síť ISP (Internet Service Provider) [2, 16].

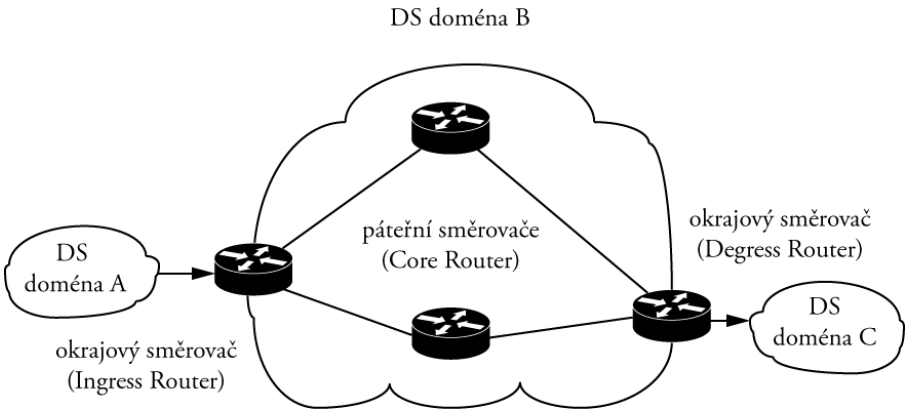
Zajištění QoS v DiffServ doméně je založeno na principu PHB (Per-Hop Behavior), tj. definovaném typu chování mezi dvěma aktivními prvky (směrovači). Každý datový paket vstupující do DiffServ domény je na její hranici klasifikován a přiřazen k toku dat s podobným chováním a poslán dále do domény. Příslušnost k tomuto toku dat se zakóduje do hlavičky IP datagramu a na základě této hodnoty (značky) a nastavené politiky služeb se pak s paketem dále pracuje [13, 16].

Pro lepší přehlednost označujeme místa na hranici DiffServ domény, kde probíhá klasifikace a značkování provozu, jako hraniční uzly (Boundary Nodes), viz obr. 2.1. Tyto uzly spojují mezi sebou různé DiffServ domény, ať už se stejnými nebo rozdílnými PHB. Síťové prvky umístěné uvnitř jednotlivých domén pak nazýváme vnitřními uzly (Interior Nodes) [2].



Obr. 2.1: Příklad označení směrovačů v DiffServ doméně

Hraniční uzly jsou charakterizovány ještě jedním parametrem, a tím je směr provozu. Pokud provoz vstupuje do domény, označujeme takový uzel jako vstupní (Ingress Node). Pokud provoz doménu opouští, označujeme jej jako výstupní (Degress Node). Budou-li dvě DiffServ domény spojeny přes jeden společný směrovač, je tento router zároveň pro jednu doménu vstupním a pro druhou doménu výstupním směrovačem [21] viz obr. 2.2.



Obr. 2.2: Označení směrovačů v DiffServ doméně z pohledu datového provozu

2.1 Funkční bloky DiffServ

V následujících kapitolách se budeme věnovat funkčním bloků, které tvoří mechanismus DiffServ. Jejich rozdělení je následující [13]:

- třídění datových jednotek,
- dohled nad provozem,
- aktivní správa front,
- řízení odesílání paketů.

2.2 Třídění datových jednotek

Prvním krokem k úspěšné implementaci DiffServ je analýza provozu, resp. klasifikace a značkování paketů. Získáme tak představu jak s pakety naložit.

2.2.1 Klasifikace paketů

Classifier (třídič [16]) provádí identifikaci přicházejících paketů a jejich následné dělení do několika skupin (tříd) podle předem definovaných pravidel.
Rozlišujeme dva druhy klasifikace:

- BA (Behaviour Aggregate),
- MF (Multi-Field Classification).

BA patří mezi nejjednodušší DiffServ třídiče [16]. Pakety jsou vybírány na základně obsahu pole DS v záhlaví IP paketu. Tento typ klasifikace se implicitně používá v případě, kdy byl paket označen dříve, než dorazil na vstupní rozhraní směrovače.

Druhým typem třídiče je MF. Ten pakety vybírá podle hodnot jednoho nebo více polí v hlavičce IP datagramu. Obvykle se používá kombinace hodnot například zdrojové adresy, cílové adresy, zdrojového a cílového portu nebo ID identifikátoru protokolu. MF třídič je flexibilnější a je možné jej použít pro složitější politiku alokace zdrojů [2, 16].

V praxi ale můžeme také narazit na problém, kdy nebudeme moci identifikovat pakety podle výše uvedených parametrů. Příkladem mohou být aplikaci pro výměnu souborů typu peer-to-peer, které dynamicky mění porty tak, aby pronikly firewally.

2.2.2 Značkování

Značkování (obarvení) je během procesu třídění datových jednotek velmi důležité. Během další cesty paketu sítí totiž určuje, jak s ním bude zacházeno.

Pokud je příchozí paket neoznačen (unmarked) tak mu tato procedura určí nastavením hodnoty pole v hlavičce IP datagramu, příslušnost ke konkrétní třídě [13, 16]. Je-li je paket už označený, ale nesplňuje kritéria daná třídou, je takováto datová jednotka vyhodnocena jako nedůvěryhodná (not trusted) a přeznačkována. Bývá obvyklé, že je takovému paketu nastavena vyšší priorita zahození v případě zahlcení sítě [4]. Jestliže je paket označený a důvěryhodný (trusted), nemusí už následující aktivní prvek provádět stejnou hloubkovou klasifikaci, ale může se věnovat jen řízení přenosu podle daného označkování [13].

Pro přenos pomocí protokolu IP ve verzi 4 (IPv4) je prováděna značkovačem (Marker) změna v poli TOS (Type Of Service) viz obr. 2.3.

Obr. 2.3: Struktura pole TOS [5, 8]

Osmibitové pole TOS má vyhrazeny první tři bity na určení typu služby a priority (IP Precedence) při zpracování směrovačem. Čím je tedy vyšší hodnota IP Precedence, tím vyšší je priorita provozu. Následující 3 bity indukují požadovaný způsob zpracování daného paketu [17]:

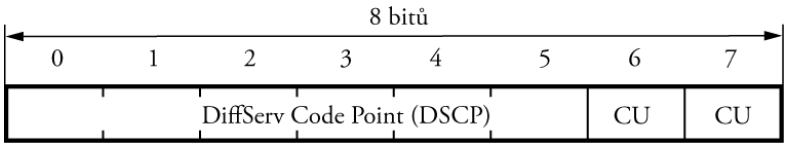
- bit D (Delay) udává požadované zpoždění (0 – normální, 1 – malé),
- bit T (Throughput) udává požadovanou propustnost (0 – normální, 1 – vysoká),
- bit R (Reliability) udává požadovanou spolehlivost (0 – normální, 1 – vysoká).

Poslední dva nulové bity, resp. s nastavenou hodnotou nula, jsou rezervovány pro případné budoucí využití (např. pro navrhovaný mechanismus řízení propustnosti [17]).

Tab. 2.1: Stupně priority IP Precedence [16]

Stupeň	IPP	Označení
0	000	Routine
1	001	Priority
2	010	Immediate
3	011	Flash
4	100	Flash Override
5	101	Critical and Emergency Call Processing
6	110	Internetwork Control
7	111	Network Control

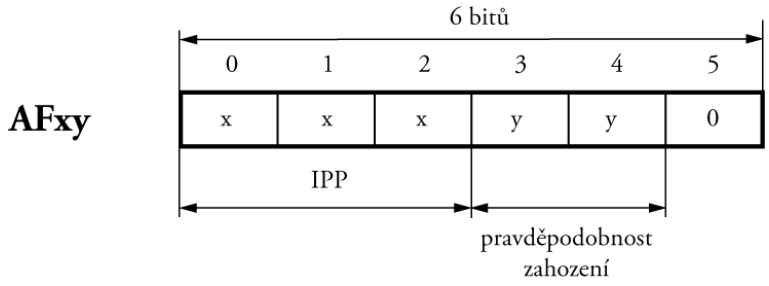
Další možností jak označkovat paket je úprava pole DS (Differentiated Service) v hlavičce IP datagramu (využití jak u IPv4, tak i IPv6), viz obr. 2.4.



Obr. 2.4: Struktura pole DS [13, 16]

Na rozdíl od pole TOS neudává absolutní prioritu paketu a způsob zpracování paketu, ale pouze identifikátor třídy. Pro hodnotu identifikátoru tak může využít DSCP (DiffServ Code Point) prvních 6 z celkových 8 bitů pole DS (Differentiated Services). To znamená, že můžeme deklarovat až 2^6, tedy 64 tříd. První 3 bity odpovídající IP Precedence zajišťují zpětnou kompatibilitu s TOS. Další dva bity určují pravděpodobnost

zahození (01 – nízká, 10 – střední, 11 – vysoká pravděpodobnost). Poslední bit DSCP je nulový. Kódovaní PHB podle DSCP je uvedeno na obr. 2.5. Zbývající dva bity pole DS jsou nulové a bývají označovány jako CU (Currently Unused) a stejně jako u pole TOS jsou určeny pro budoucí využití [13, 16].



Obr. 2.5: Kódovací schéma PHB

Tab. 2.2: Doporučené hodnoty IP Precedence s odpovídajícími třídami PHB [16]

Stupeň	Hodnoty DSCP a odpovídající třídy PHB								
	Nízká pravděpodobnost zahození			Střední pravděpodobnost zahození			Vysoká pravděpodobnost zahození		
0	BE	000000	0	-			-		
1	AF11	001010	10	AF12	001100	12	AF13	001110	14
2	AF21	010010	18	AF22	010100	20	AF23	010110	22
3	AF31	011010	26	AF32	011100	28	AF33	011110	30
4	AF41	100010	34	AF42	100100	36	AF43	100110	38
5	EF	101110	46	-			-		
6	-			-			-		
7	-			-			-		

2.3 Dohled nad provozem

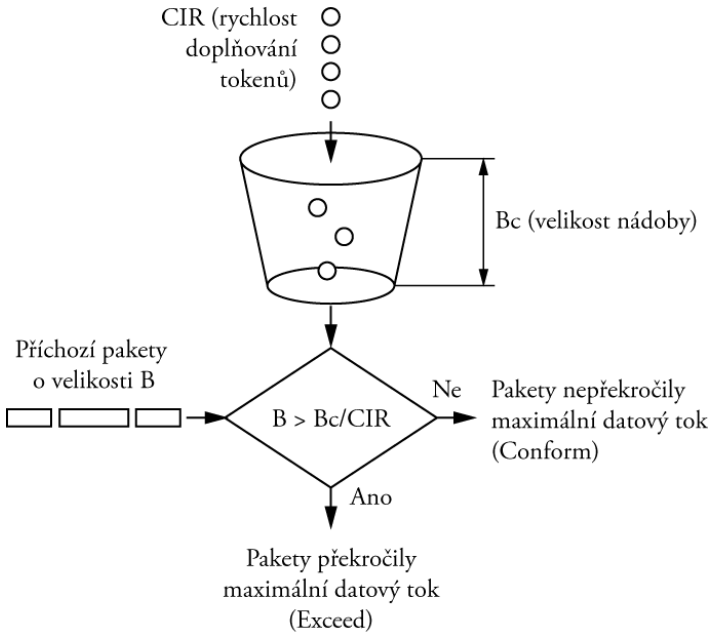
Dohled nad provozem, neboli správa šířky pásma (Traffic Rate Management), patří k těm nejstarším ze skupiny mechanismů QoS [12]. Je to proces sledování síťového provozu vybraného během klasifikace a výsledky jsou porovnávány s profilem definovaným pro daný datový tok. Tímto profilem označujeme dohodnuté podmínky, podle kterých jsou odpovídající pakety směrovány na výstupní rozhraní [16]. Obě

používané metody Policing a Shaping mají stejný cíl – nastavit maximální datový tok, který daný síťový přenos nemůže překročit. K jeho dosažení používají podobné metody, odlišným způsobem však na danou situaci reagují.

2.3.1 Policing

Tato metoda a její nástroje, tzv. Policers, omezují provoz tak, že pakety překračující pásmo (tzv. data out-of-contract) jednoduše zahodí nebo jim případně přeznačkují TOS pole v IP hlavičce [22].

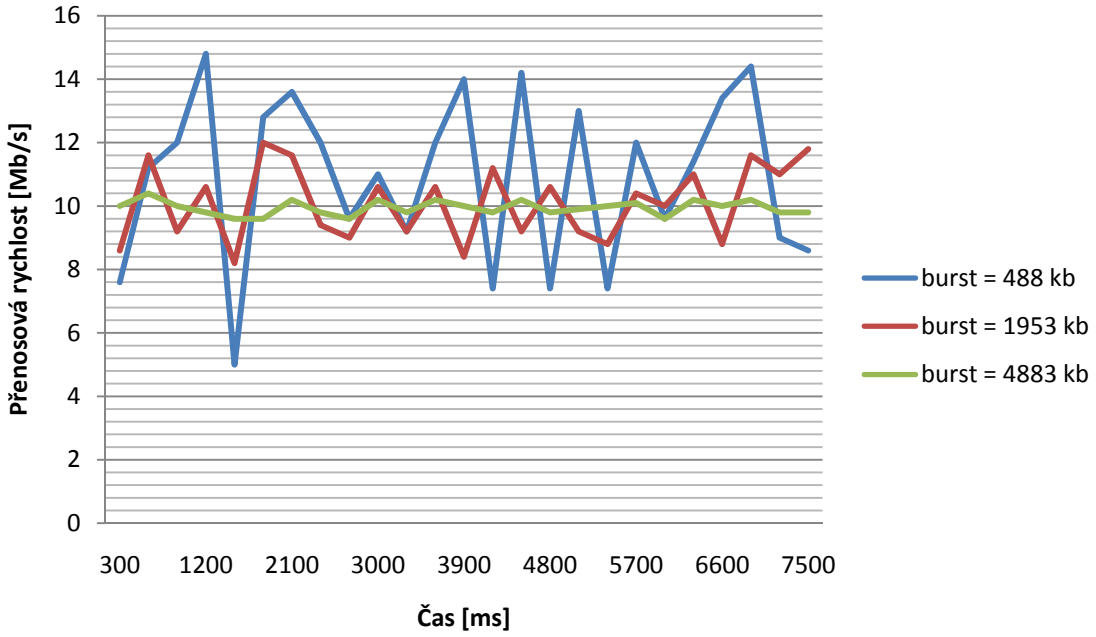
Pro výpočet množství dat, které mohou být odeslány bez toho, aby došlo k překročení kapacity, je nejčastěji používán [16] algoritmus Token Bucket, viz obr. 2.6.



Obr. 2.6: Princip činnosti algoritmu Token Bucket [13, 21]

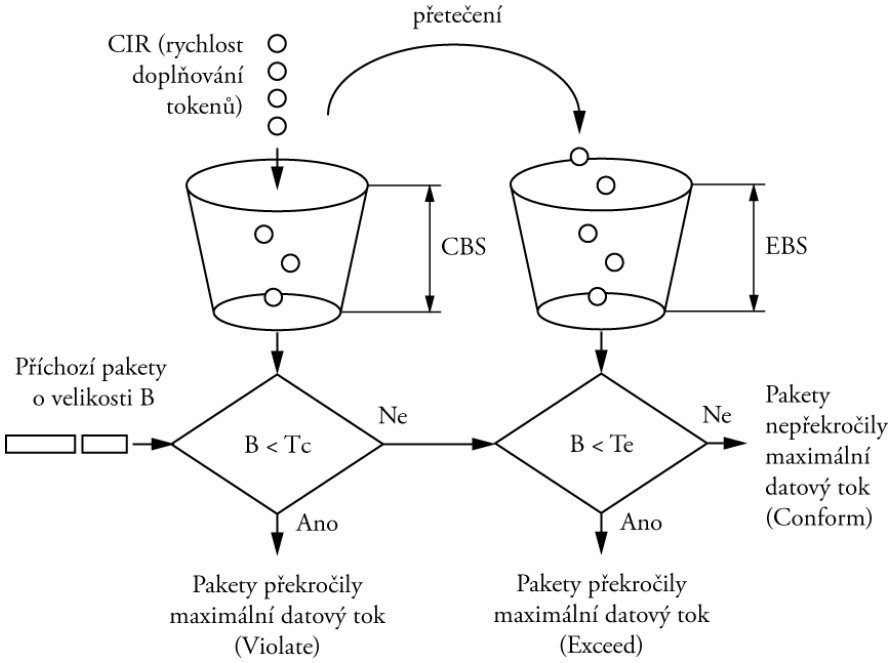
Pokud se budeme držet názvu, jde o kbelík (Bucket), ve kterém je obsaženo určité množství kupónů (Token). Každý jednotlivý kupón představuje povolení k odeslání určitého množství bytů do sítě (1 byte paketu odpovídá 1 kupónu). Pokud tedy chceme odeslat paket, musíme z kbelíku odebrat odpovídající počet kupónů. Pokud potřebné množství kupónů k dispozici nemáme, paket musí počkat nebo se zahodí. Kupóny jsou

do kbelíku doplňovány zadanou rychlostí (definovanou CIR – Commited Information Rate) až do chvíle, kdy se naplní do definované velikosti (Bc – Burst Size). Ukázka jak ovlivňuje velikost Bc síťový provoz je na obr. 2.7.



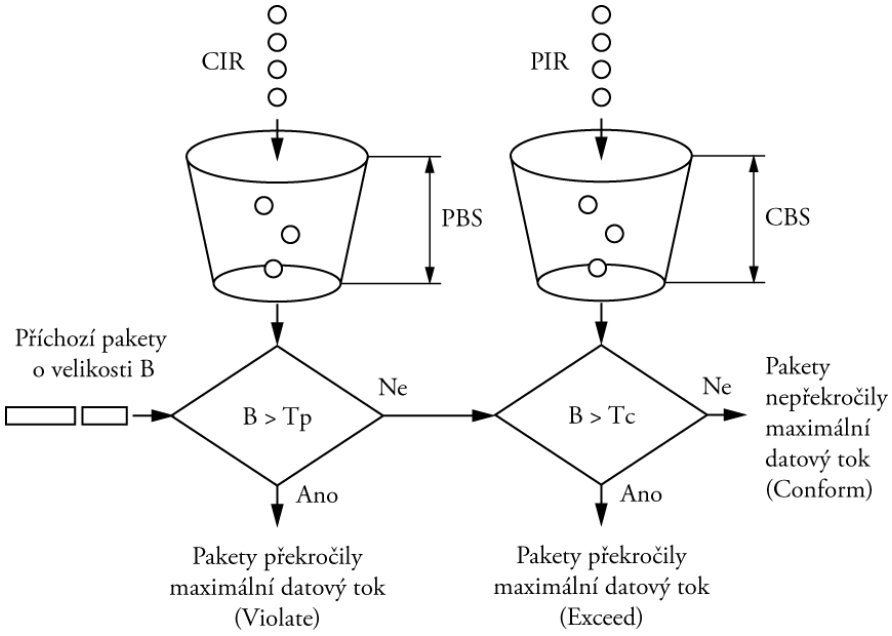
Obr. 2.7: Příklad tří datových toků s různě nastavenou velikostí Bc [1]

Kromě výše popsané verze, označované jako Single Token Bucket (Single-Rate Two-Color Policer), existuje ještě Dual Token Bucket algoritmus (Single-Rate Three-Color Policer) viz obr. 2.8.



Obr. 2.8: Princip činnosti algoritmu Dual Token Bucket [20]

Základem této verze je, že máme k dispozici dva kbelíky. Pokud tedy není při příchodu paketu dostatek kupónů v prvním, podíváme se do druhého. V případě, že v něm máme dostatek kupónů, paket se odešle, v opačném případě se zahodí. Podstatný rozdíl mezi algoritmy je ale v tom, jakým způsobem se druhý kbelík kupóny naplňuje. První způsob naplňování nastává v případě, že první kbelík je plný a kupóny místo zahození tak v podstatě „přepadnou“ do druhého kbelíku [5, 21].

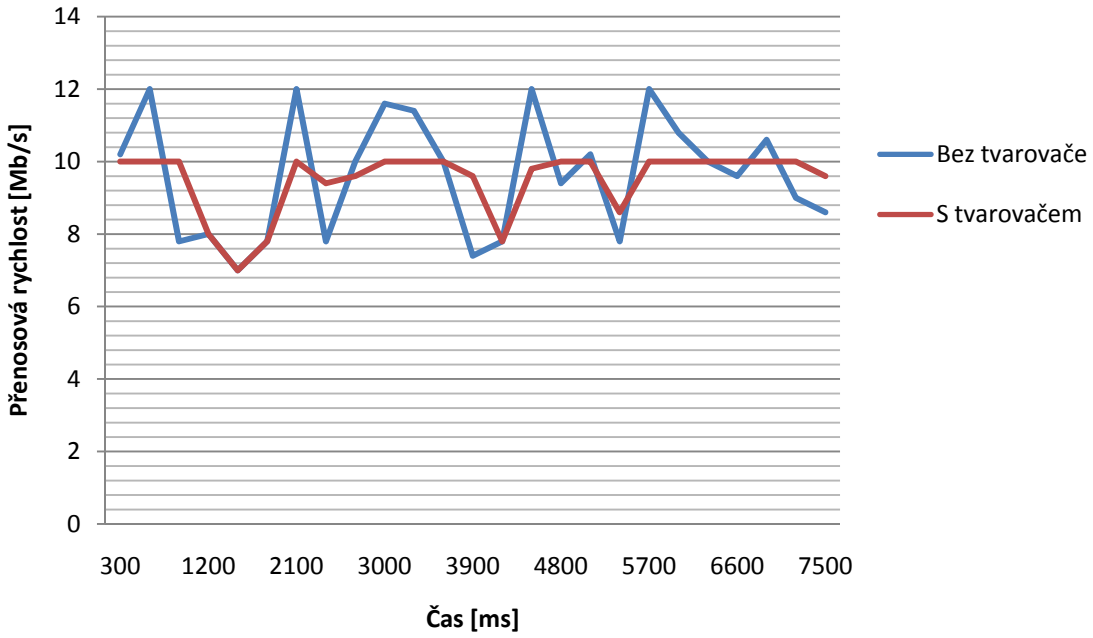


Obr. 2.9: Princip algoritmu Dual-Rate Token Bucket [20]

Druhou možností (tzv. Dual-Rate Three-Color Policer) je, že se kupóny přidávají jednou rychlostí do prvního a druhou rychlostí do druhého kbelíku, viz obr. 2.9. Při konfiguraci potom musíme nastavit nejen CIR (Commited Access Rate) ale i PIR (Peak Access Rate) [5].

2.3.2 Shaping

Tvarovač má za úkol upravit příchozí pakety tak, aby odpovídaly danému datovému profilu. Jinými slovy brání průchodu paketům do té doby, dokud se datový tok datovému profilu nepřizpůsobí [16]. Princip je zobrazen na obr. 2.10.



Obr. 2.10: Ukázka vlivu tvarovače na síťový provoz [1]

Pro měření využívají nástroje Shaper stejně jako Policer algoritmus Token Bucket. Shaper ale na rozdíl od Policeru využívá možnosti Token Bucketu k vyhlazení toku dat. Obecně je nastaven Token Bucket na Cisco prvcích s Bc (Commited Burst) na CIR/8 z čehož vyplývá, že Tc (Time Interval) je 125ms. Tato hodnota je vyhovující pro běžný datový provoz. Při nasazení realtimových aplikací už ale způsobuje zbytečné zpoždění [20].

Aby tvarovače plnily správně svoji funkci, přidávají k Token Bucketu ještě další mechanismy. Nejstarším mechanismem implementovaným do Cisco IOS je GTS (Generic Traffic Shaping). Přicházející pakety zařazuje do bufferu a v pravidelných intervalech je odesílá s omezenou přenosovou rychlostí. Modernějším způsobem je pak využití fronty pomocí mechanismu CBWFQ (Class-Based Weighted Fair Queuing). Přicházející pakety jsou řazeny do fronty, ze které se pak odebírají do té doby, dokud jsou k dispozici volné tokeny. Pokud volné tokeny nejsou, s pakety se nic neděje, jednoduše počkají ve frontě na další kupóny [12, 22].

Konfigurace shapingu na směrovačích Cisco je znatelně méně komplikovaná než nasazování policingu. Nejpoužívanější mechanismus CBWFQ nám pomocí mapy třídy identifikuje tok a mapa politiky pak upřesňuje hodnoty CIR, Bc, aj. Experimentovat v podstatě můžeme pouze s těmito parametry, ale optimální hodnoty jsou nastaveny samotným výrobcem [5, 12].

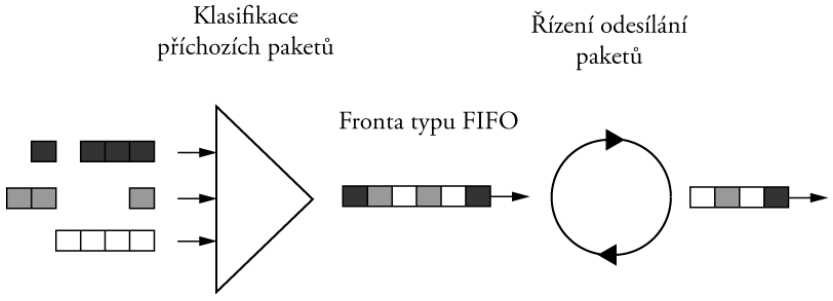
2.4 Řízené odesílání paketů

Po roztřídění provozu jsou jednotlivé označkované pakety zpracovávány mechanismy front. Správné zpracování paketů do front je velmi důležité a má na efektivitě DiffServ zásadní podíl. Tuto problematiku označujeme také jako správa zahlcení (Congestion Management). Základní metody řízení odesílání paketů [13] jsou fronty typu FIFO (First-In First-Out), PQ (Priority Queuing), WFQ (Weighted Fair Queuing), CBWFQ (Class-Based Weighted Fair Queuing), LLQ (Low Latency Queuing).

V praxi je to s implementací mechanismů front u Cisca složitější, protože nabídka a možnosti mechanismů závisí na použitém zařízení. Velké rozdíly jsou nejen mezi přepínači a směrovači, ale i mezi samotnými směrovači, kdy vyšší řady směrovačů používají k řízení odesílání paketů speciální hardware.

2.4.1 FIFO

Jde o nejjednodušší základní mechanismus se zachováním pořadí požadavků. Po uvolnění linky se postupně odesílají pakety podle pořadí příchodu, viz obr. 2.11. Bohužel tento typ front nám se zahlcením příliš nepomůže. Neumí zvýhodnit určité pakety a nezabezpečí tak ochranu proti aplikacím zahlcujícím síť. Další nevýhoda nastává ve chvíli, kdy je fronta plná. Přicházející pakety jsou potom okamžitě zahazovány [13].

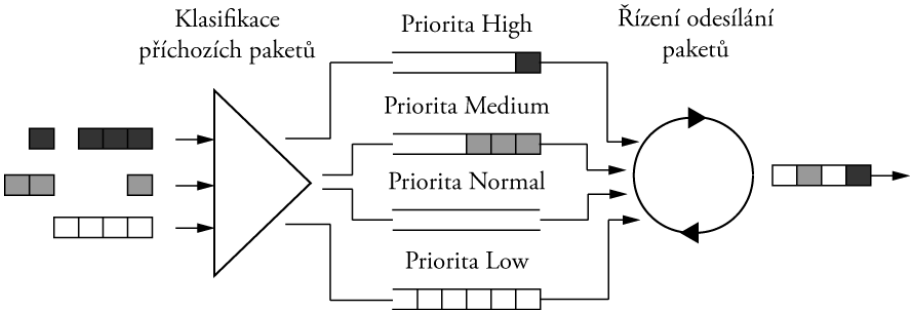


Obr. 2.11: Princip fungování fronty typu FIFO [13]

Typické využití najdeme u všech hardwarových rozhraní aktivních prvků. Další využití je pro TCP/UDP a i když byla fronta FIFO tím prvním krokem ke kontrole datového provozu, podmínky dnešních sítí vyžadují mnohem sofistikovanější mechanismy [13].

2.4.2 PQ

Fronta s prioritní obsluhou (PQ) je při zachování jednoduchosti fronty typu FIFO schopna přidělit jednotlivým frontám určitou prioritu, viz obr. 2.12.



Obr. 2.12: Princip fungování fronty PQ [13]

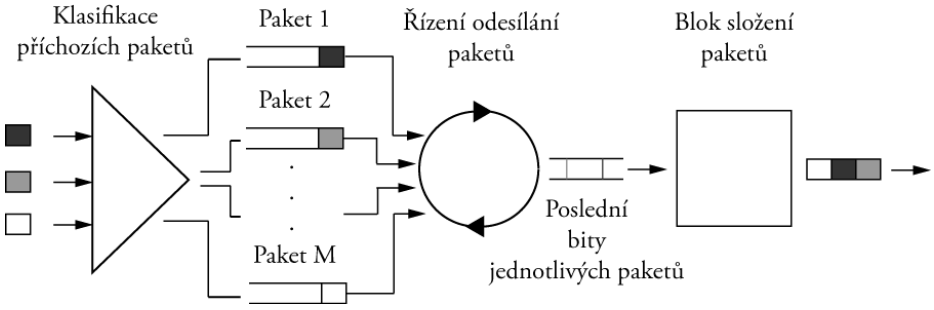
Při příchodu je paket podle přiřazené priority zařazen do jedné z front, např. High, Medium, Normal a Low. Pakety bez přiřazené priority se automaticky v tomto případě zařazují do fronty Normal. Pokud se cesta uvolní, nejprve se odesílají data z front s nejvyšší prioritou, teprve poté přicházejí postupně na řadu fronty s nižší prioritou.

Na první pohled je zřejmé, že hlavním problémem PQ bude „uvíznutí“ paketů ve frontách s nízkou prioritou. Než se na ně dostane řada, nemají žádnou šanci nějak svoje pakety odeslat. Pro vyšší vrstvy se pak tváří tyto pakety díky velkému zpoždění jako ztracené [13, 15]. Pomocí PQ můžeme pohodlně realizovat EF (Expedited Forwarding) PHB. Postačí nám k tomu dvě fronty, jednu přiřadíme EF a druhou pro službu Best-Effort [10].

2.4.3 WFQ (Weighted Fair Queuing)

Mechanismus WFQ pracuje s prioritou (neboli váhou) paketu. Rozhodnutí o tom, které pakety se zahodí a které se zpracují přednostně, je založené na počtu bitů v poli IP Precedence. WFQ zajišťuje, aby pakety s vyšší prioritou byly v případě zahlcení zahozeny s menší pravděpodobností. K této činnosti využívá vlastnosti statického algoritmu s lineárním výsledkem, viz obr. 2.13. Pravděpodobnost zahození paketu s prioritou 5 je pětkrát nižší než u paketu s prioritou 0. Pro provoz s pakety o vyšší prioritě vyhrazuje algoritmus vyšší objem šířky pásma, čímž zajišťuje rychlejší obsluhu datového toku i v případě zahlcení sítě [22].

Kromě toho jistou váhu přiřazuje i jednotlivým datovým tokům. Tím určuje pořadí vysílaných paketů umístěných ve frontě. Nejdříve systém obsluhuje pakety s nižší váhou. Protože je priorita dělitelem váhového faktoru, dostane síťový provoz s prioritou číslo 7 menší váhu než provoz s hodnotou pole IP Precedence 2. Má tedy v pořadí pro vysílání přednost [22].



Obr. 2.13: Princip fungování fronty WFQ [11]

Ve výsledku WFQ tak zajišťuje, že datové provozy s nízkým objemem (což je obvykle převládající provoz) mají při odesílání paketů přednost. Naopak datovým tokům s velkým objemem nezbývá nic jiného, než se o zbývající kapacitu šířky pásma rozdělit [22].

WFQ je navržena s ohledem na nejběžnější typ síťového provozu. Minimalizuje počet a nutnost konfiguračních činností a automaticky se dynamicky přizpůsobuje podmínkám v síťovém provozu. Efektivně využívá celou šířku pásma. Tyto vlastnosti

umožnily, aby WFQ byla standardně přednastavována na všech sériových rozhraních v síťových prvcích Cisco [13].

2.4.4 CBWFQ (Class-Based Weighted Fair Queuing)

Jde o hybridní spojení vlastností WFQ a CQ (Custom Queuing). Je sice složitější než zmiňované techniky, ale při řízení odesílání paketů spojuje do sebe to nejlepší z obou mechanismů. Algoritmus CBWFQ řadí pakety do front podle tříd. Třídy provozu pak určují parametry, jako je příslušné vstupní rozhraní, seznamy řízení přístupu nebo použité síťové protokoly. Pro každou z těchto nadefinovaných tříd se vytvoří fronta, do které se pakety posílají podle své příslušné třídy. K třídám přiřadíme charakteristické vlastnosti, například šířku pásma, váhu, hodnotu maximálního množství paketů ve frontě (maximální délku fronty) [12, 13, 22].
S použitím CBWFQ získáváme dvě důležité možnosti: přidělovat šířku pásma a velice flexibilně upravovat podobu třídy podle zavedených kritérií.

2.4.5 LLQ (Low-Latency Queuing)

LLQ v sobě spojuje vlastnosti PQ a CBWFQ. Vytváří striktně prioritní fronty s nízkou latencí a garantovanou šířkou pásma. Zároveň jsou hlídány tak, aby v případě zahlcení nepřekročily zadanou šířku pásma. Technika pomocí vlastností CBWFQ umožňuje na úrovni třídy vytvořit jednu prioritní frontu [12].

2.5 Aktivní správa front

Oproti mechanismům v předešlé kapitole, které mají za úkol zvládnout nastalé zahlcení sítě, mají nástroje této oblasti DiffServ (Congestion Avoidance) za úkol blížícímu se zahlcení předcházet [22]. Detekci zahlcení a krizový stav před zaplněním front zvládají pomocí inteligentního zahazování paketů [13].

Pokud pomineme nejjednodušší mechanismus zahazování paketů Tail Drop, kdy je využíván princip fronty typu FIFO [12], používáme k předcházení zahlcení dvě základní metody [13]:

- nepřímé řízení propustnosti pomocí mechanismů RED (Random Early Detection), WRED (Weighted Random Early Detection) a další,
- přímé řízení propustnosti pomocí ECN (Explicit Congestion Notification).

2.5.1 RED (Random Early Detection)

Princip mechanismu RED je založen na předem aplikovaných opatřeních, která mají zabránit možnému zahlcení datového toku TCP. Návrh mechanismu RED představili v roce 1990 Sally Floyd a Van Jacobson. Hlavní myšlenkou RED je neustálé sledování front, a v případě hrozícího zahlcení, vybere mechanismus náhodný TCP datový toky a začne pakety zahazovat. Čím více se fronta zaplní, tím více je vybráno datových toků a zahozeno větší množství paketů. Aby se situace neustále neopakovala, využívá RED vlastnosti TCP, kdy v reakci na zahazování paketů odešle směrovač ke zdroji o této události zprávu a ten reaguje zpomalením vysílání na polovinu. Po určité době bez zpráv o zahazování paketů vrátí rychlost vysílání opět na původní úroveň [22].

V praxi pak máme možnost nastavit rychlost zahazování a prahovou hodnotu, při které se začínají pakety zahazovat [22].

2.5.2 WRED (Weighted Random Early Detection)

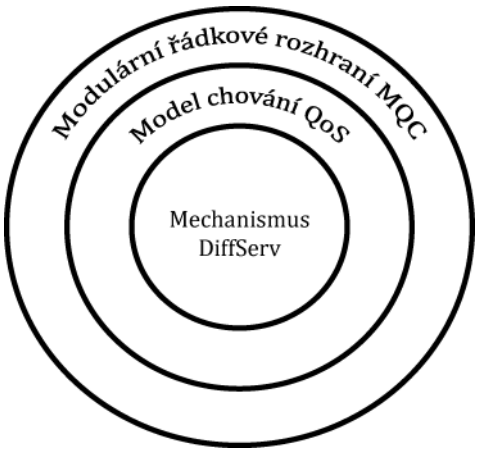
I když RED představoval ve své době velký krok k aktivní správě front, dnes jej Cisco nedoporučuje. Místo něj nabízí svou pokročilejší implementaci WRED. Stejně jako u metody RED i zde dochází k zahazování paketů ještě před zahlcením sítě. Pakety jsou tedy zahazovány dříve, než dojde k naplnění vyrovnávací paměti (buffer). Rozdíl je v tom, jaké pakety jsou zahazovány. Algoritmus WRED pakety s vyšší prioritou zahazuje s nižší pravděpodobností než pakety s nižší prioritou [13].

Mechanismus WRED je nejčastěji používán v páteřních směrovačích, kam přicházejí z hraničních směrovačů pakety s přiřazenou prioritou.

3. IMPLEMENTACE QOS NA SÍŤOVÝCH PRVCÍCH CISCO

Firma Cisco se už od svého vzniku nezaměřuje v síťových technologiích pouze na hardwarovou stránku prvků, ale také na software umožňující jejich řízení a konfiguraci. Předmětem práce je nasazení QoS na směrovačích Cisco, takže všechny následující popisy mechanismů a konfigurací se budou týkat výhradně jich.

Firma Cisco založila svou implementaci kvality služeb na obecném modelu DiffServ. Ten tvoří pomyslný základní kámen. O něj se opírá samotná implementace, která je Ciscem označovaná jako QoS Behavioral Model, tedy model chování QoS. Ten věrně simuluje nástroje a v podstatě celý proces zpracování provozu mechanismu DiffServ. Nad tím vším je poslední, vrchní vrstva, konfigurační framework tzv. modulární řádkové rozhraní MQC (Modular QoS Command-Line Interface) [1]. Vztah mezi DiffServ, modelem chování a MQC je zobrazen níže na obr. 3.1.



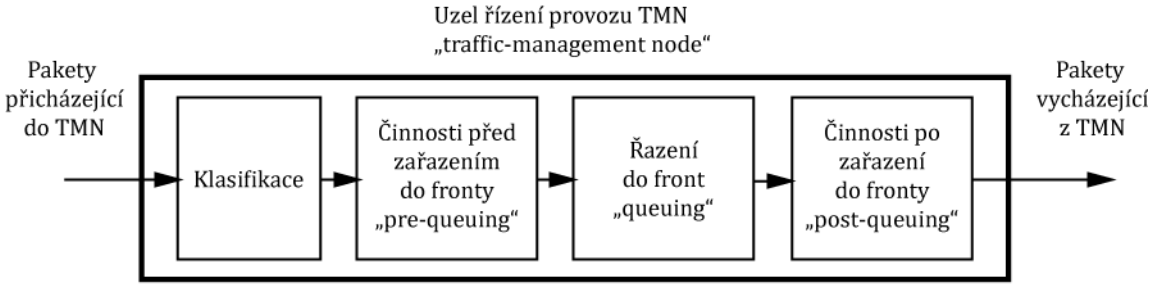
Obr. 3.1: Cisco implementace QoS nad mechanismem DiffServ [1]

MQC společně s vrstvou modelu chování QoS skrývají před uživatelem jednotlivé technicky složité detaily komplexní implementace [1]. Veškerou konfiguraci QoS pak uživatel provádí běžnými příkazy, které jsou známé z Cisco IOS, pomocí příkazového řádku CLI (Command-Line Interface).

3.1 Model chování QoS

Druhou vrstvu implementace tvoří model chování QoS. Vytváří jakousi jednotnou platformu pro MQC, která obhospodařuje implementace QoS na jednotlivých typech zařízení, ať už se jedná o různé modelové řady směrovačů, přepínačů apod.

Model chování QoS je založen na funkci entity, nazvané uzel řízení provozu (TMN - Traffic-Management Node). Ten představuje souhrn akcí, které zařízení aplikuje v určitém bodě na provoz při průchodu paketu. TMN identifikuje jeden nebo více datových toků a určí, která akce se na jaký tok aplikuje. Pomáhají mu v tom jeho čtyři komponenty (obr. 3.2). Jejich pořadí kopíruje zpracování provozu u DiffServ a všechny lze konfigurovat [1].



Obr. 3.2: Blokové schéma TMN [1]

V bloku klasifikace se identifikuje přicházející datový tok. Nejčastěji podle záhlaví paketu, ale možností je mnohem více, jak bude ukázáno dále. Uzel poté přiřadí ke každému typu datového toku název třídy. Pokud datový tok nesplňuje žádná stanovená kritéria klasifikace, je automaticky přiřazen ke třídě class-default. V případě, že TMN blok klasifikaci neobsahuje, je do class-default přiřazen celý příchozí datový provoz. Blok klasifikace tedy přijímá příchozí síťový provoz, identifikuje jednotlivé datové toky a přiřadí jim název třídy [1].

Druhým blokem je tzv. pre-queuing, což je skupina QoS činností, která vždy následuje po klasifikaci a předchází řazení do front (pokud jsou fronty použity). Umožňuje např. provést značkování (marking), měření (policing), zahození paketu (dropping), kompresi záhlaví paketu aj. Činnosti se stejně jako u DiffServ vzájemně ovlivňují, např. měřič může způsobit přeznačkování, ale nikdy neovlivní výsledek

klasifikace. To znamená, že v TMN probíhá klasifikace vždy jen jednou a to v klasifikátoru [1].

Řazení do front (queuing) umožňuje správu šířky pásma v případě zahlcení. Jde v podstatě o paměť, kam se pakety ukládají [6]. Pokud TMN frontu neobsahuje, pakety se při zahlcení rovnou zahazují. Řazení do front vždy následuje až po klasifikaci a bloku činností pre-queuing a předchází bloku činností post-queuing. Skládá se ze dvou částí: zařazení do fronty (enqueuing) a vyřazení z fronty (dequeuing). Blok zařazení řadí pakety do fronty. Pomocí mechanismů Tail drop, RED a WRED kontroluje velikost fronty. Blok vyřazení naopak pomocí mechanismů omezování provozu (shaper) řídí odchod paketů z fronty [1, 6].

Poslední komponentou TMN je blok činností po řazení do front. Jedná se o činnosti prováděné před tím, než paket opustí uzel. Typickým příkladem takových činností jsou různé kompresní mechanismy [1].

3.2 MQC – modulární řádkové rozhraní

Modulární rozhraní MQC bylo na prvopočátku vyvíjeno jako framework pro model činností QoS, umožňující vyžívat CBWFQ. Časem se však vyvinulo v komplexní nástroj pro zajištění kvality služeb. Díky modelu chování QoS, který se stará o implementaci na konkrétní hardware, si zachovalo svoji modulárnost. Další zásadní koncepce, která vydržela už od dob návrhu, je třídění provozu na základě tříd. Proto se také v souvislosti s MQC používá výraz Class-based QoS [16].

Během konfigurace MQC tedy postupujeme podle těchto tří kroků [4]:

- definování mapy tříd (traffic class definition),
- definování mapy politik (policy definition),
- aplikování mapy politik na rozhraní (policy application).

Nejprve se provoz klasifikuje jednou nebo více mapami tříd. Mapy tříd se uplatní na mapy politik a mapy politik se použijí na rozhraní jako politiky služby.

3.2.1 Definování mapy tříd

V definicích tříd provozu vytváříme pojmenované třídy. Do těchto tříd vybíráme, nebo pomocí těchto tříd, volíme pro nás zajímavý provoz, se kterým hodláme dále pracovat. Díky jménu třídy je následné zacházení s takto vybraným provozem relativně jednoduché [16].

Vytvoříme si tedy mapu tříd (maximální počet tříd je 64) pomocí příkazu class-map, jejímž názvem může být libovolný textový řetězec:

```
ROUTER(config)#class-map nase-testovaci-trida
```

Příchozí paket musí vyhovovat všem podmínkám a pak o něm lze říci, že je paketem dané třídy:

```
ROUTER(config)#class-map match-all
```

Může nastat situace, kdy budeme potřebovat, aby paket náležel dané třídě v případě shody s alespoň jednou libovolnou podmínkou:

```
ROUTER(config)#class-map match-any
```

Pokud bychom potřebovali uplatnit politiky QoS na veškerý provoz na rozhraní, musíme vytvořit mapu, které vyhovují všechny pakety:

```
ROUTER(config-cmap)#match any
```

Filtrovat provoz můžeme podle řady parametrů, např. podle typu protokolu, podle shody s přístupovým seznamem (paket vyhovuje podmínce v případě, že je zadán v přístupovém seznamu v tzv. ACL – Access Control List), podle shody s cílovou MAC adresou (paket vyhovuje podmínce, pokud je určen pro cílovou MAC adresu):

```
ROUTER(config-cmap)#match protokol ip
ROUTER(config-cmap)#match access group 30
ROUTER(config-cmap)#match destination-address mac 00-16-FA-E1-E3-88
```

Existuje samozřejmě celá řada dalších podmínek. Dokonce máme možnost nejen testovat na shodu (jak jsme si výše ukázali), ale i na neshodu s podmínkou. Mapy tříd můžeme také zanořovat do sebe pro složitější výběry.

3.2.2 Definování mapy politik

V prvním kroku jsme vybrali provoz, který nás zajímá, a nyní vytvořením mapy politik nastavíme politiku. V tomto kroku můžeme uplatnit všechny nástroje, které nám QoS nabízí. Můžeme označit provoz, provést značkování, vyhradit šířku pásma pro provoz, apod.

Nejprve tedy definujme mapu politik. Názvem může být opět libovolný textový řetězec:

```
ROUTER(config)#policy-map nase-testovaci-politika
```

Potom určíme, které mapě tříd budeme nastavovat vlastnosti (ty se pak budou uplatňovat na celý provoz klasifikovaný do dané třídy):

```
ROUTER(config-pmap)#class nase-testovaci-trida
```

Nyní už můžeme přistoupit k řadě nastavení, která nám QoS poskytuje. Např. nastavení hodnoty DSCP protokolu IP (hodnoty 0 až 63), nastavení IP priority (v rozmezí hodnot 0 až 7, přičemž nejvyšší hodnotě odpovídá nejvyšší priorita), přidělení přenosové kapacity dané třídě (nastavujeme pomocí WFQ, v procentech nebo kb/s):

```
ROUTER(config-pmap-c)#set dscp 46
ROUTER(config-pmap-c)#set ip precedence 5
ROUTER(config-pmap-c)#set bandwidth 1024
```

Stejně jako u map tříd, i zde existuje ještě mnoho jiných nastavení. Důležité ale je, že provoz, který jsme nezařadili do žádné třídy, se automaticky řadí do tzv. defaultní třídy (class-default), na kterou se nevztahují politiky QoS. Není ovšem problém v případě potřeby její chování změnit, například velikostí fronty paketů:

```
ROUTER(config)#policy-map politika-defaultni-tridy
ROUTER(config-pmap)#class class-default
ROUTER(config-pmap-c)#queue-limit 8
```

3.2.3 Aplikování mapy politik na rozhraní

Posledním krokem k nadefinování QoS politiky je její přiřazení k rozhraní, v rámci kterého se bude provádět. Důležité je také určit, zda se nastavení QoS bude týkat příchozích, nebo odchozích paketů. Příklad ukazuje aplikaci nase-testovaci-politika na rozhraní GigabitEthernet1/0/1 v příchozím směru:

```
ROUTER(config)#interface GigabitEthernet1/0/1
ROUTER(config-if)#service-policy input nase-testovaci-politika
```

3.3 Další nástroje Cisco pro QoS

3.3.1 AutoQo

Další možností jak implementovat QoS u síťových prvků Cisco je AutoQoS. Jedná se o uživatelsky jednoduchý nástroj, který umožňuje několika příkazy nakonfigurovat na zařízení doporučené QoS nastavení. K dispozici je ve dvou verzích: VoIP (Voice over IP) a Enterprise. AutoQoS VoIP je verze určena především k zabezpečení přenosu hlasového provozu. AutoQoS Enterprise využívá model deseti tříd, podle kterých klasifikuje data, a tyto třídy obsluhuje [12].

AutoQoS je určen pro administrátory sítí, kteří nemají čas nebo možnosti věnovat se hlouběji principům a mechanismům QoS. Pokud vyžadujeme flexibilnější a sofistikovanější řešení, nezbývá než začít pracovat s mnohem komplexnějším MQC CLI (Modular QoS Command-Line Interface).

4. SCÉNÁŘE MĚŘENÍ

K měření parametrů QoS můžeme přistupovat různými způsoby. Prvním je tzv. aktivní měření (někdy označované jako trasování). Při této metodě pro účely měření generujeme "umělý" datový tok. Nevýhodou je, že jde o čistě "laboratorní" měření, na druhou stranu máme plnou kontrolu nad tím, jak datový tok vypadá. Tato metoda je obzvlášť vhodná k testování konkrétního vlivu vybraného mechanismu. Další možností je připojit se k už existující síti a sledovat probíhající "živý" datový přenos. Tato metoda je reálnému prostředí mnohem blíže, ale musíme dbát na to, aby samotné měření ovlivnilo výsledek co nejméně. Nakonec můžeme obě metody zkombinovat. V takovém případě pak generovaný provoz putuje jednotlivými uzly sítě společně s původním síťovým provozem. Data pak získáme jak sledováním informací z generovaných dat, tak sledování změn v původním síťovém provozu způsobených přidáním tohoto generovaného toku.

Ideálním postupem je pomocí aktivní metody prozkoumat vliv jednotlivých mechanismů QoS na parametry sítě, vybrat nejvhodnější mechanismy a aplikovat je v reálném provozu a dále pokračovat v sledování přínosu konkrétního zajištění QoS v reálných podmínkách.

Prvořadým cílem je tedy prozkoumat vliv nastavení QoS na zpoždění, kolísání zpoždění a ztrátovost paketů. Kromě těchto naměřených veličin pak můžeme pokračovat v testování zajištění kvality služeb s ohledem na subjektivní kvalitativní parametry uživatele. Jde v podstatě o vnímání spokojenosti s poskytovanou službou, což se týká především služeb přenosu audia a videa v reálném čase.

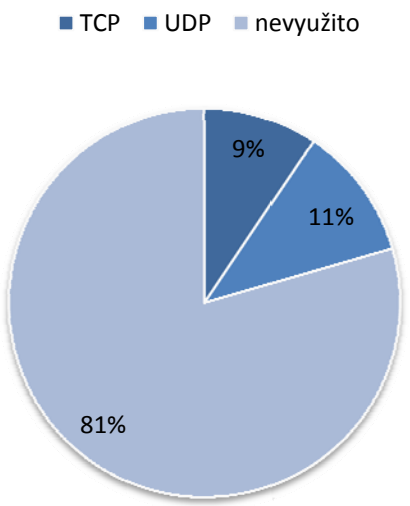
Při měření a testech jednotlivých mechanismů QoS v této práci bude postupováno podle dvou scénářů měření. Linka bude postupně zatěžována stále větším objemem dat. Síťový provoz budou tvořit vždy stejně definovaná hlasová data (protokol UDP) a ostatní data představující zátěž (protokol TCP). V jednotlivých scénářích se liší pouze počty těchto dvou typů generovaných toků.

4.1 Měření s nižším zatížením

V tomto scénáři tvořilo datový tok deset současně probíhajících přenosů hlasových dat (UDP) a dva toky ostatních dat (TCP). Konkrétní parametry jednotlivých datových toků jsou uvedeny v tab. 4.1. Průměrná rychlost datového toku v tomto scénáři dosahuje 2101 kb/s. Rozložení jednotlivých typů provozu v tomto scénáři je na obr. 4.1.

Tab. 4.1: Parametry jednotlivých generovaných toků

Datový tok	Protokol	Rychlost odesílání paketů	Velikost paketů	Průměrná rychlost
Hlasová data	UDP	100 p/s	120 B	960,1 kb/s
Ostatní data	TCP	81 p/s	840 B	1140 kb/s



Obr. 4.1: Rozložení jednotlivých typů dat vůči celkové šířce pásma

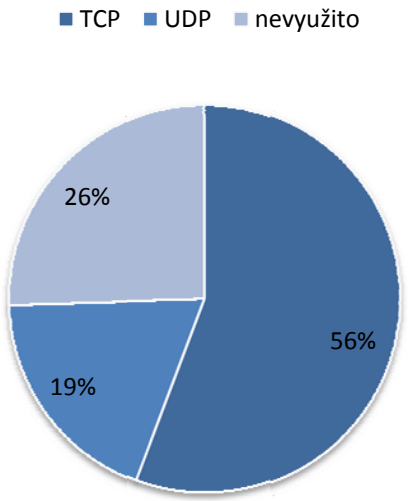
4.2 Měření s vyšším zatížením

V tomto scénáři tvořilo datový tok dvacet současně probíhající přenosů hlasových dat (UDP) a 10 toků ostatních dat (TCP). Konkrétní parametry jednotlivých datových toků jsou uvedeny v tab. 4.2. Průměrná rychlost datového toku v tomto scénáři

dosahuje 7,624 kb/s a končí tak na hranici 75% propustnosti linky. Tato hranice je doporučována firmou Cisco. Rozložení jednotlivých typů dat je na obr. 4.2

Tab. 4.2: Parametry jednotlivých generovaných toků

Datový tok	Protokol	Rychlost odesílání paketů	Velikost paketů	Průměrná rychlost
Hlasová data	UDP	100 p/s	120 B	1920 kb/s
Ostatní data	TCP	81 p/s	840 B	5702 kb/s



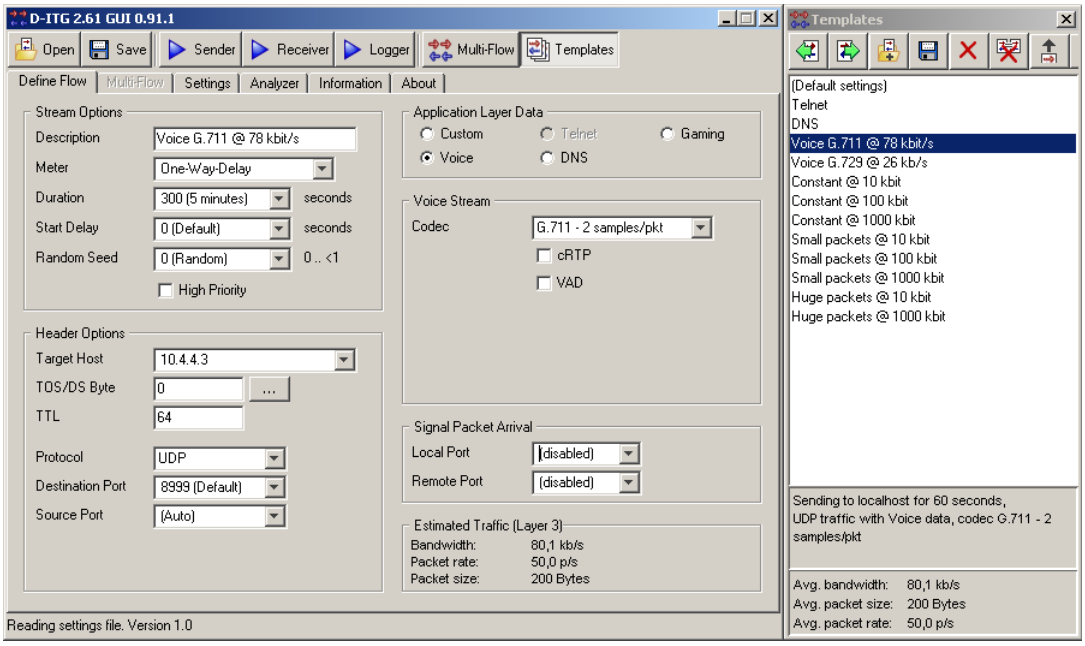
Obr. 4.2: Rozložení jednotlivých typů dat u vyššího zatížení vůči celkové šířce pásma

4.3 Použitý software

4.3.1 Generování dat

Pro generování datového toku jsem se rozhodl použít D-ITG (Distribued Internet Traffic Generator). Dokáže vygenerovat datové pakety typu TCP, UDP, ICMP, DNS, Telnet a VoIP a podporuje protokoly IPv4 a IPv6. Pro generovaný tok byl použit přednastavený profil pro VoIP paket a čistý paket protokolu TCP. Kromě Linuxu existuje i verze pro operační systém Windows s přehledným grafickým rozhraním. Další obrovskou výhodou programu je možnost generování a zároveň zachytávání datového toku.

Z důvodu větší přehlednosti bylo pro konfiguraci parametrů jednotlivých toků využito grafické rozhraní D-ITG 2.6 GUI 0.91.1 beta. Ukázka konfigurace datového toku je zobrazena na obrázku 4.3.



Obr. 4.3: Ukázka možnosti nastavení parametrů datového toku pomocí GUI rozhraní

Výstupem grafického rozhraní byl soubor s definovanými parametry generovaných toků, který byl následně využit v konzolovém rozhraní D-ITG.

Generování dat probíhalo v režimu OWD (One Way Delay). Výsledkem byl vygenerovaný log soubor formátu dat, jaký používá například generátor MGEN. Soubor byl dekódován do textových a csv souborů. Údaje ze souborů byly poté zpracovány pomocí programů MATLAB a Microsoft Excel.

Protože je program D-ITG citlivý na synchronizaci času mezi vysílačem a přijímačem (v případě špatné synchronizace vychází značně zkreslené hodnoty), byly před každým měřením počítače (počítač generující data a počítač přijímající data) časově synchronizovány vůči NTP serveru v síti Internet.

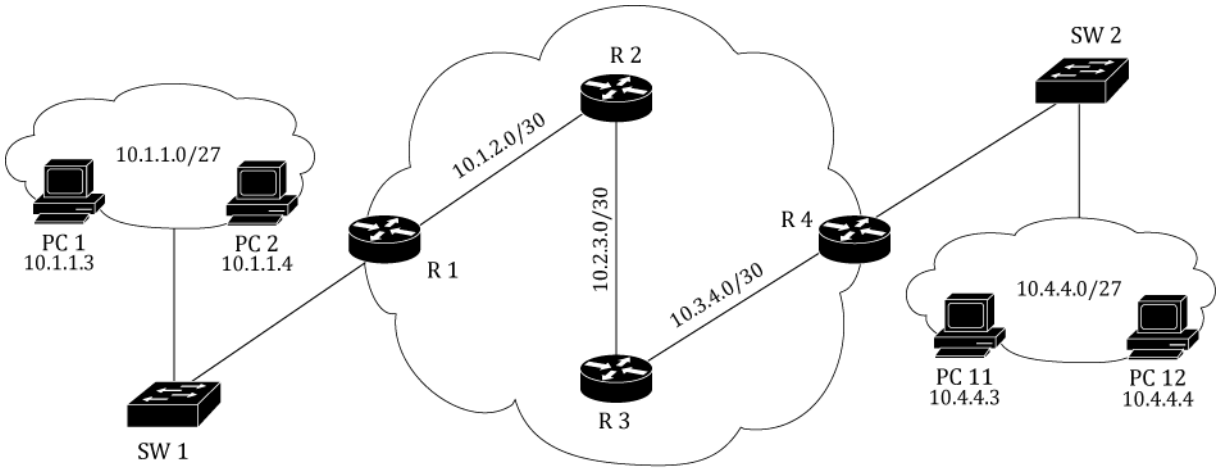
4.3.2 Monitorování provozu

K monitorování provozu na rozhraních směrovačů byl použit program Cisco SDM (Security Device Manager). Provoz na rozhraních počítačů byl kontrolován pomocí programu SoftPerfect Network Protocol Analyzer. Kontrolu použité klasifikace obstarával program Wireshark.

5. MĚŘENÍ MECHANISMŮ ŘÍZENÉHO ODESÍLÁNÍ PAKETŮ A AKTIVNÍ SPRÁVY FRONT

5.1 Topologie testovací sítě

Zapojení testovací sítě je znázorněno níže na obr. 5.1. Jádrem sítě jsou čtyři směrovače tvořící DiffServ doménu. Okrajové směrovače R1 a R4 tvoří tzv. hranici důvěry (Trust Boundary) a provádějí klasifikaci a značkování provozu přicházejícího z nedůvěryhodného prostředí. Páteřní směrovače R2 a R3 pak na tomto datovém provozu klasifikaci již neprovádějí a pouze aplikují mechanismy zajištění kvality služeb.



Obr. 5.1: Zapojení testovací sítě

5.1.1 Směrovače

Použitými směrovači (R1 až R4) byly Cisco 1841. Tyto směrovače patří do výrobní řady 1800, která je součástí rodiny Cisco Integrated Services Router (ISR). Použité směrovače byly tzv. modulární, což znamená, že umožňují instalaci přídavných modulů k rozšíření podporovaných funkcí. Tyto směrovače se nejčastěji používají k připojení k WAN (Wide Area Network) sítím nebo při zabezpečení VPN (Virtual Private Network) spojení.

Původní návrh sítě předpokládal propojení směrovačů pomocí sériového rozhraní rychlostí 2 Mb/s. Bohužel během měření byl vlivem přepětí zničen napájecí obvod směrovače R2 i se sériovým rozhraním. Vzhledem ke vzniklému zdržení, způsobenému úspěšnou opravou napájecího obvodu, zároveň ale neúspěšným pokusům zprovoznit sériové rozhraní, padlo rozhodnutí propojit všechny směrovače pomocí FastEthernetových portů. Jde společně s konzolovým a AUX vstupem o základní výbavu celé modelové řady 1841. Pro účely měření byla rychlost portů snížena ze 100 Mb/s na 10 Mb/s.

5.1.2 Přepínače

Úlohu přepínačů v testovací síti obstarávaly dva přepínače Cisco Catalyst 2950. První přepínač s 48 FastEthernetovými porty je označen SW1 a druhý s 24 FastEthernetovými porty SW2.

5.1.3 Pracovní stanice

Pracovní stanice tvořily 4 notebooky značek Fujitsu Siemens, Hewlett-Packard a Acer. Všechny obsahovaly nainstalovaný systém MS Windows XP. U dvou notebooků byl v průběhu testování vyzkoušen také operační systém CentOS pro porovnání výsledků měření pod různými operačními systémy.

5.2 Výběr mechanismů řízeného odesílání paketů

V první části měření budou otestovány implementace mechanismů řízeného odesílání paketů: FIFO, PQ, CBWFQ a LLQ. FIFO je na rozhraních standardně nastaveno výrobcem. Měření prioritních front bude probíhat na mechanismech PQ a LLQ. Mechanismus PQ je vybrán v podstatě jen kvůli možnosti porovnání, Cisco jej dnes kvůli jeho nedostatkům nedoporučuje a pomalu přestává být podporován. Oproti tomu LLQ je jeden z těch novějších přírůstků do rodiny podporovaných mechanismů QoS

implementovaných firmou Cisco. Z dalších mechanismů rozebraných v kapitole 2 nebyl do měření zahrnut mechanismus WFQ, Cisco jej nedoporučuje používat na linkách rychlejších jak 2 Mb/s. Proto byl pro testovací síť s nastavenou rychlostí 10 Mb/s jako zástupce vážených front otestován mechanismus CBWFQ.

5.3 Měření mechanismu FIFO (bez zajištění QoS)

Úlohou prvního provedeného měření bylo zjištění hodnot jednotlivých parametrů ovlivňujících kvalitu služeb bez jakéhokoliv konfigurování jejích mechanismů. Z tohoto důvodu byl kromě nastavení propustnosti všech použitých rozhraní zprovozněn pouze směrovací protokol OSPF (Open Shortest Path First).

5.3.1 Konfigurace

Protože testovací síť byla propojena pomocí FastEthernetových rozhraní, na kterých je od výrobce přednastaven mechanismus front na FIFO, nebyla potřeba žádná konfigurace.

Kontrola konfigurace aplikované na rozhraní:

```
R1#show interface FastEthernet0/0
FastEthernet0/1 is up, line protocol is up
  Hardware is Gt96k FE, address is 001b.d4f0.b45d (bia 001b.d4f0.b45d)
  Description: SMER SW1
  Internet address is 10.1.1.1/27
  MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Full-duplex, 10Mb/s, 100BaseTX/FX
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:00, output 00:00:00, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 1000 bits/sec, 2 packets/sec
  5 minute output rate 1000 bits/sec, 2 packets/sec
     1599341 packets input, 601438567 bytes
     Received 1022 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 watchdog
     0 input packets with dribble condition detected
     240970 packets output, 16780410 bytes, 0 underruns
     0 output errors, 0 collisions, 3 interface resets
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
```

5.3.2 Měření s nižší zátěží

Pro měření s nižší zátěží byly použity datové toky popsané v kapitole 4.1. Nasazení tohoto typu generované zátěže by síti ani aktivním prvkům nemělo činit potíže. Výsledky měření jsou uvedeny v tabulkách 5.2 a 5.3.

Tab. 5.2: Naměřené hodnoty sledovaných parametrů přenosu hlasových dat

Sledovaný parametr	Naměřená hodnota
Průměrná hodnota zpoždění	13,79 ms
Maximální hodnota zpoždění	55,50 ms
Průměrná hodnota kolísání zpoždění	0,472 ms
Ztrátovost paketů	0%

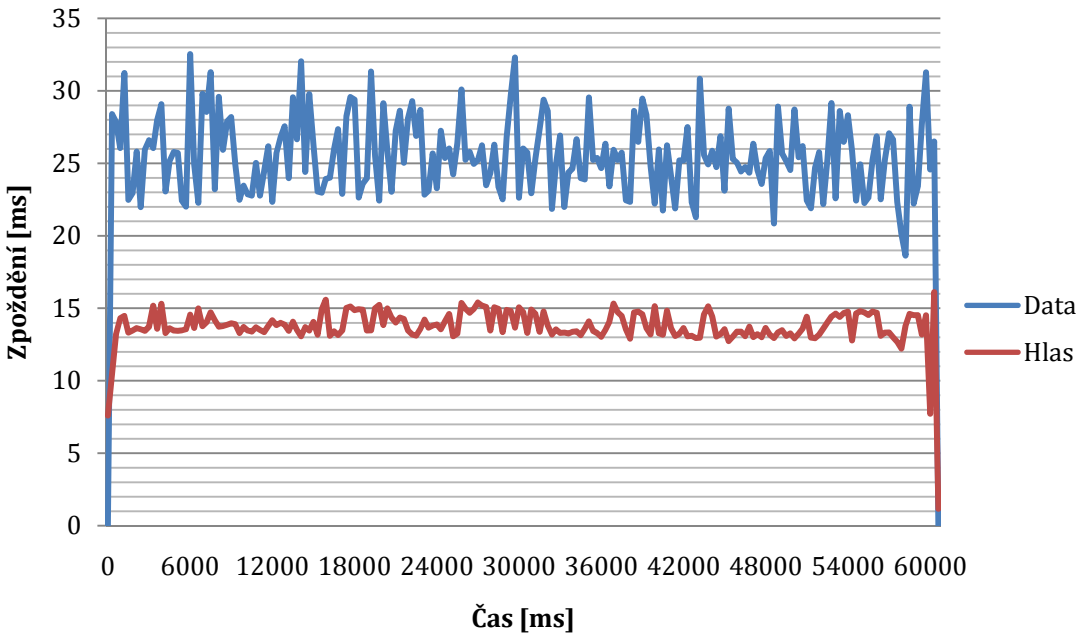
Tab. 5.3: Naměřené hodnoty sledovaných parametrů přenosu ostatních dat

Sledovaný parametr	Naměřená hodnota
Průměrná rychlost	544,7 kb/s
Průměrná hodnota zpoždění	25,42 ms
Průměrná hodnota kolísání zpoždění	8,599 ms
Přenesených dat	100%

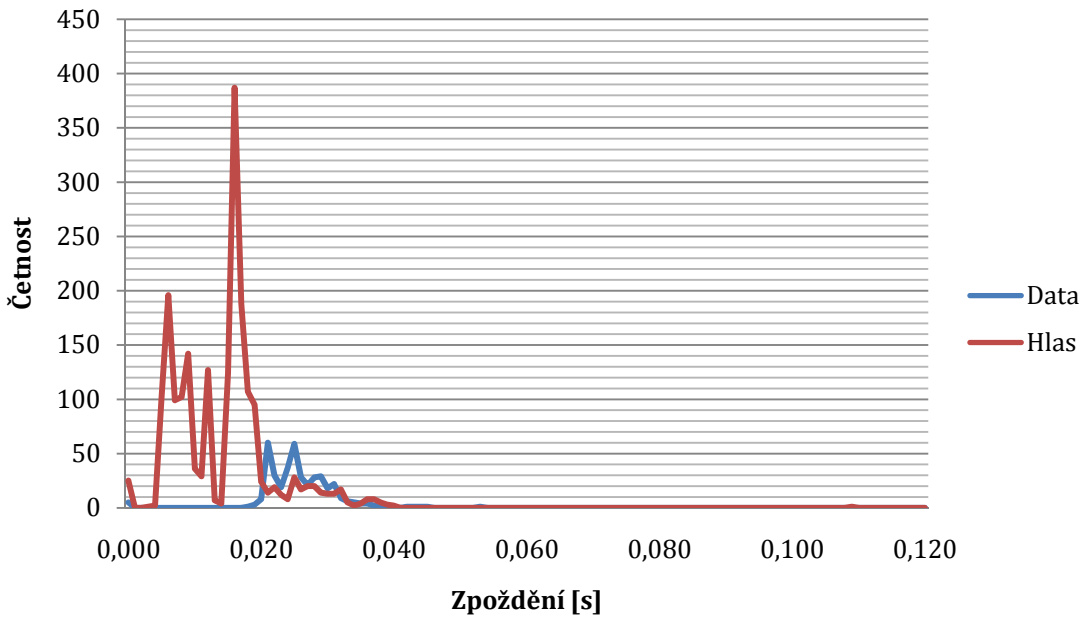
Jak je vidět z naměřených hodnot, úvodní předpoklad o bezproblémovém zvládnutí generovaného provozu se potvrdil. Datové toky s protokolem TCP vykazovaly 0% ztrátovosti paketů. U mnohem náročnějších hlasových dat byly splněny podmínky stanovené ITU (International Telecommunication Union) G.114:

- ztrátovost paketů do 1%,
- hodnoty kolísání zpoždění do 30 ms,
- hodnoty zpoždění do 150 ms.

Obr. 5.2 znázorňuje časový průběh zpoždění pro hlasová i ostatní data. Společně s grafem na obr. 5.3 je vidět, že všechny hodnoty zpoždění se pohybují v rozmezí 0-150 ms. Společně s 0% ztrátovosti paketů by byl takový hlasový provoz bezproblémový.

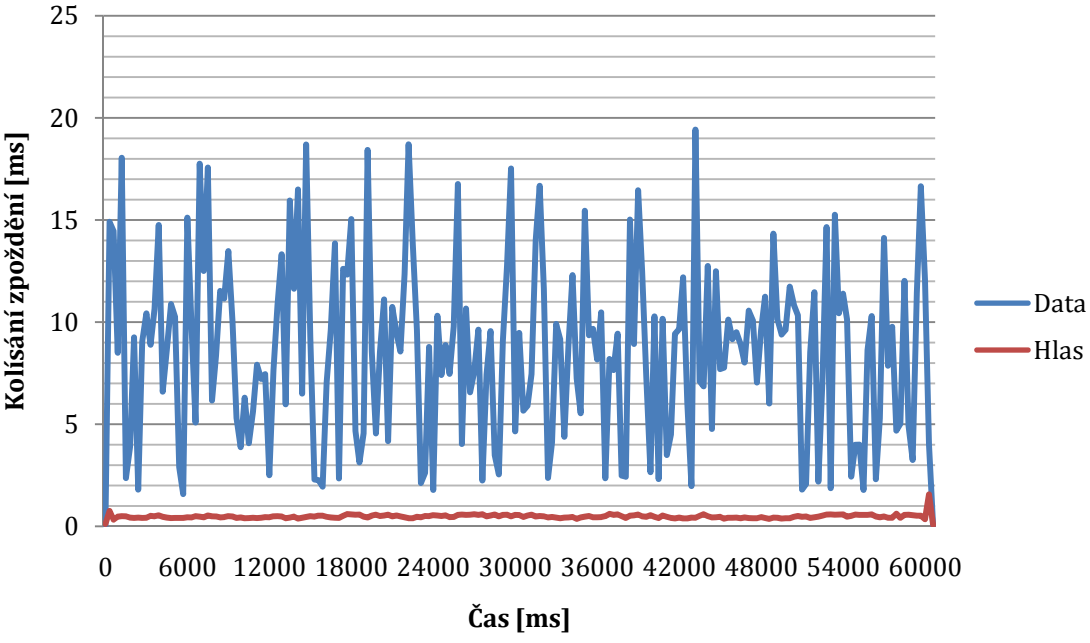


Obr. 5.2: Průběh zpoždění hlasových a ostatních dat



Obr. 5.3: Histogram zpoždění hlasových a ostatních dat

Rovněž kolísání zpoždění, jehož časový průběh pro oba toky dat je vidět v grafu na obr. 5.4, splňuje u hlasových dat podmínky ITU G.114.



Obr. 5.4: Průběh kolísání zpoždění hlasových a ostatních dat

5.3.3 Měření s vyšší zátěží

Při této zátěži je očekáváno zhoršení parametrů přenosu, především u hlasových dat. Se zvyšující se zátěží by měly být čím dál více patrné nevýhody plynoucí ze stejného přístupu ke všem typům provozu. Výsledné hodnoty sledovaných veličin jsou uvedeny v tabulkách 5.4 a 5.5.

Tab. 5.4: Naměřené hodnoty sledovaných parametrů přenosu hlasových dat

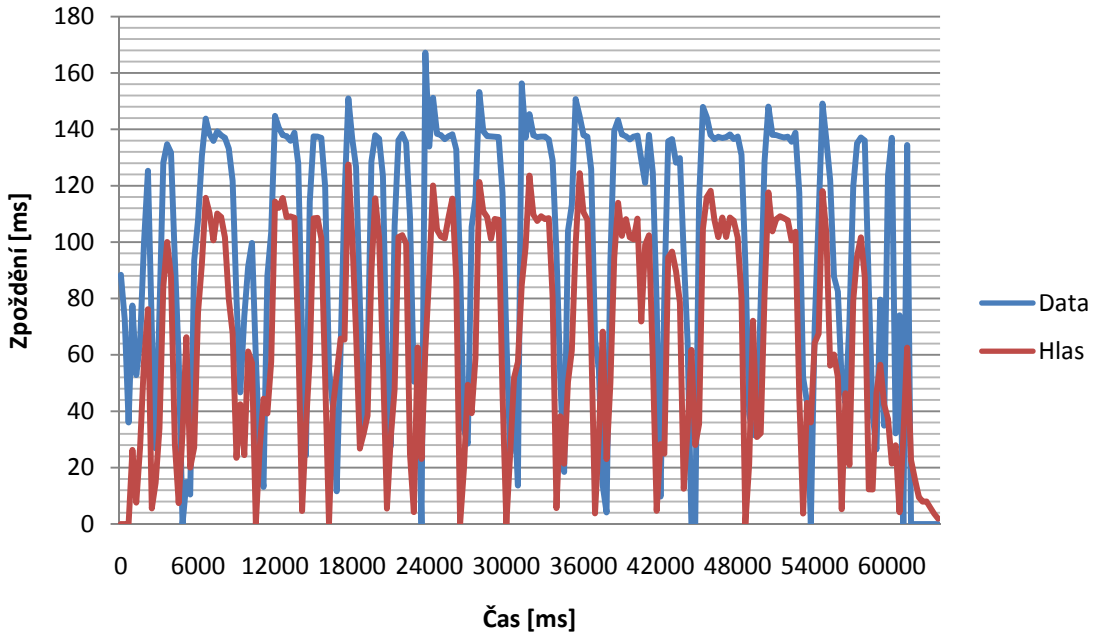
Sledovaný parametr	Naměřená hodnota
Průměrná hodnota zpoždění	122,3 ms
Maximální hodnota zpoždění	1043 ms
Průměrná hodnota kolísání zpoždění	1,314 ms
Ztrátovost paketů	0,3%

Tab. 5.5: Naměřené hodnoty sledovaných parametrů přenosu ostatních dat

Sledovaný parametr	Naměřená hodnota
Průměrná rychlost	539,1 kb/s
Průměrná hodnota zpoždění	136,3 ms
Průměrná hodnota kolísání zpoždění	13,47 ms
Přenesených dat	100%

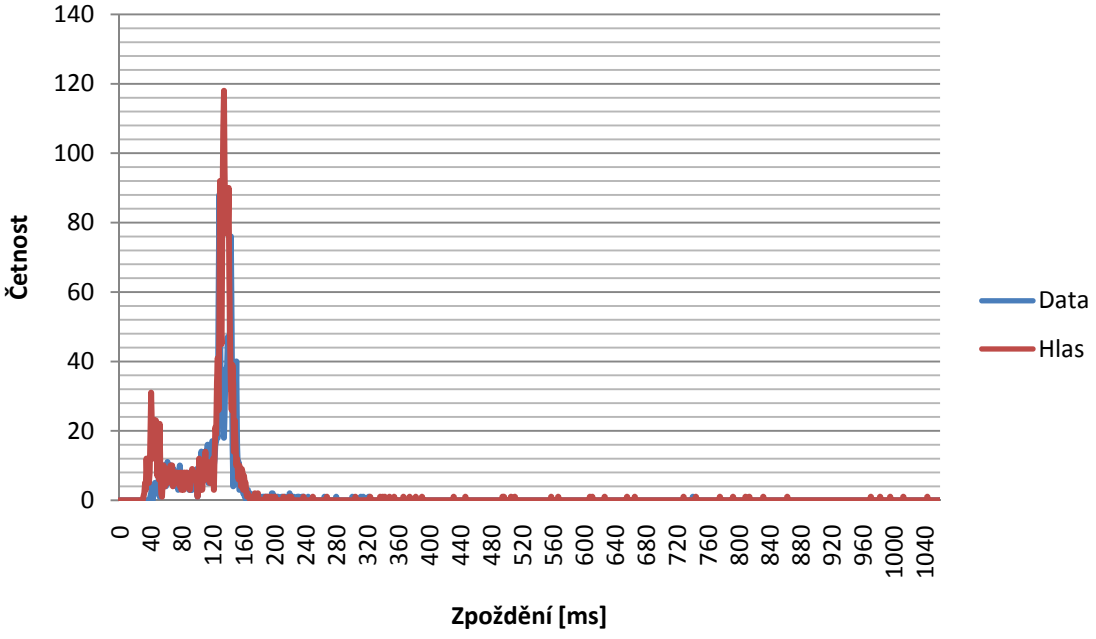
Výsledky sledovaných parametrů přenosů dat úvodní předpoklad naplňují. Prudce vzrostly hodnoty zpoždění, které se nyní pohybují v horní části intervalu 0–150 ms. Více jak 16% procent hodnot zpoždění se pohybuje v intervalu 150-400 ms, který podle ITU G.114 představuje již degradovanou službu. Jak je vidět v tab. 5.4, maximální hodnota zpoždění dokonce překračuje i tento interval a s dalšími 2% naměřených hodnot tak již představuje neakceptovatelné zpoždění. Při těchto parametrech je provoz hlasových služeb více než problematický.

Z grafu na obr. 5.5 lze vysledovat nárůst hodnot zpoždění jak u hlasových tak i ostatních dat.



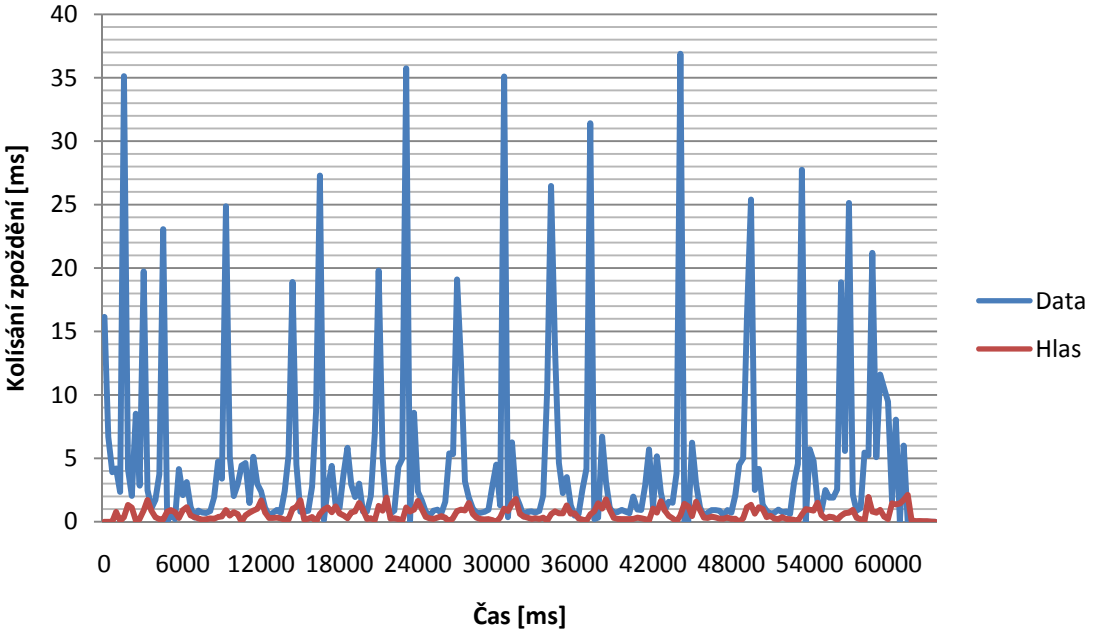
Obr. 5.5: Průběh zpoždění hlasových a ostatních dat

Histogram na obr. 5.6 ukazuje četnost jednotlivých hodnot zpoždění. Největší četnost vykazují hodnoty zpoždění mezi 0-150 ms.



Obr. 5.6: Histogram zpoždění hlasových a ostatních dat

Stejně jako narůstaly hodnoty zpoždění, vzrostly společně s vyšším provozem i hodnoty kolísání zpoždění. Přesto ještě hodnoty u hlasových dat nedosahují maximální akceptovatelné hodnoty 30 ms. Časový průběh kolísání zpoždění během měření je zobrazen na obr. 5.7.



Obr. 5.7: Průběh kolísání zpoždění hlasových a ostatních dat

5.3.4 Zhodnocení

Výsledky všech provedených měření dopadly podle očekávání. Čím více se zvětšovala zátěž generovanými daty, tím více se zhoršovaly jednotlivé parametry přenosu. Obzvláště je to patrné na přenosu hlasových dat, kde začalo docházet nejen k nárůstu hodnot zpoždění a kolísání zpoždění, ale i ke ztrátám paketů. Kvalitní přenos hlasových dat je s těmito parametry bez nasazení QoS vyloučen.

5.4 Mechanismus PQ

Hlavním úkolem mechanismu PQ (Priority Queueing) je absolutní prioritizace vybraného typu dat. Při testování této techniky řazení do front byl využit fakt, že jsou generovány dva druhy provozu. Data s protokolem UDP byla zařazena do fronty high,

tedy fronty s nejvyšší prioritou obsluhy. Ostatní data s protokolem TCP byla zařazena do prioritně nejnižší fronty low. Vzhledem k výše uvedenému se nepředpokládají problémy s přenosem hlasových dat.

5.4.1 Konfigurace

Konfigurace mechanismu představuje tři kroky: vytvoření přístupového seznamu, vytvoření prioritního seznamu a aplikovaní na rozhraní.

Vytvoření přístupového seznamu, který vybere provoz s UDP pakety:

```
R1(config)#access-list 101 permit udp any any
```

Druhým krokem je vytvoření prioritního seznamu. Prioritní řazení je nastaveno tak, že pakety UDP vybrané přístupovým seznamem budou zařazeny do fronty high, ostatní provoz bude zařazen do fronty low:

```
R1(config)#priority-list 1 protocol ip high list 101
R1(config)#priority-list 1 default low
```

Kromě přiřazení priority provozu můžeme také nastavit maximální velikost front. Následující příklad nastavuje frontě high délku 100 paketů, frontě medium 40 paketů, frontě normal 60 paketů a frontě low 200 paketů:

```
R1(config)#priority-list 1 queue-limit 100 40 60 200
```

Posledním krokem je aplikování na rozhraní:

```
R1(config-if)#priority-group 1
```

Kontrola konfigurace aplikované na rozhraní:

```
FastEthernet0/0 is up, line protocol is up
  Hardware is Gt96k FE, address is 001b.d4f0.b45c (bia 001b.d4f0.b45c)
  Description: SMER R2
  Internet address is 10.1.2.1/30
  MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec,
     reliability 255/255, txload 15/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Full-duplex, 10Mb/s, 100BaseTX/FX
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:04, output 00:00:07, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: priority-list 1
  Output queue (queue priority: size/max/drops):
     high: 0/100/0, medium: 0/40/0, normal: 0/60/0, low: 0/200/0
  5 minute input rate 0 bits/sec, 12 packets/sec
  5 minute output rate 606000 bits/sec, 86 packets/sec
     224775 packets input, 15301602 bytes
     Received 1424 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 watchdog
     0 input packets with dribble condition detected
     690949 packets output, 508850340 bytes, 0 underruns
     0 output errors, 0 collisions, 4 interface resets
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
```

5.4.2 Měření s nižší zátěží

Pro toto měření byl využit scénář měření s nižším zatížením. Generováno bylo 10 toků hlasových dat a 2 toky ostatních dat. S přihlédnutím k principu fungování prioritní fronty jsou předpokládány u hlasových dat nižší hodnoty jak u zpoždění, tak i kolísání zpoždění oproti ostatním datům. V tabulkách 5.6 a 5.7 jsou shrnuty výsledky měření pro sledované parametry přenosů hlasových a ostatních data.

Tab. 5.6: Naměřené hodnoty sledovaných parametrů přenosů hlasových dat

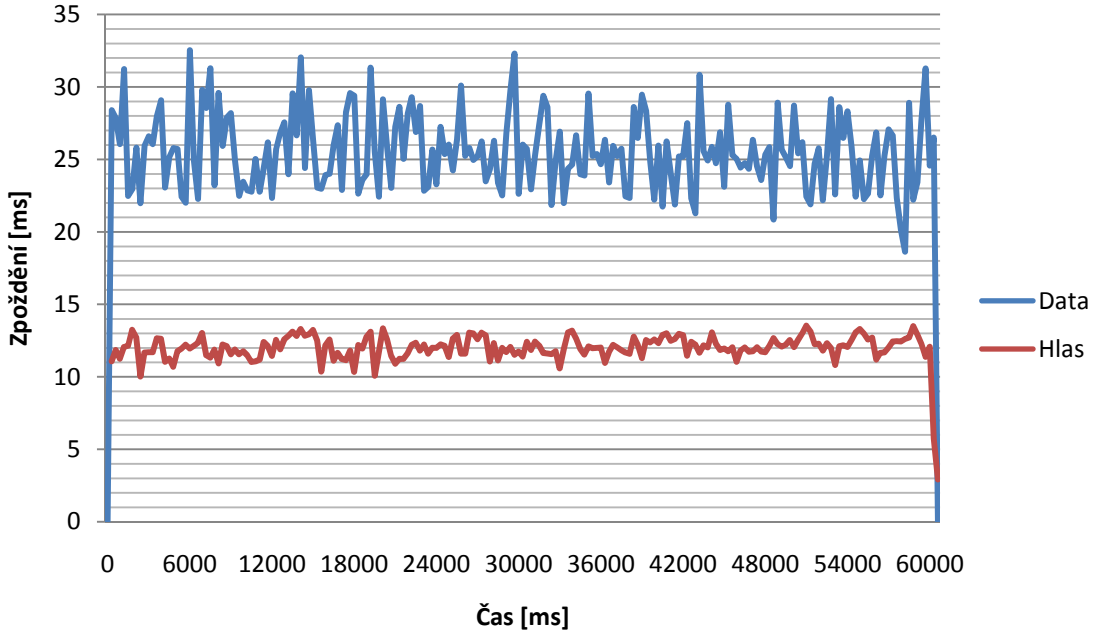
Sledovaný parametr	Naměřená hodnota
Průměrná hodnota zpoždění	11,48 ms
Maximální hodnota zpoždění	16,60 ms
Průměrná hodnota kolísání zpoždění	0,459 ms
Ztrátovost paketů	0%

Tab. 5.7: Naměřené hodnoty sledovaných parametrů přenosu ostatních dat

Sledovaný parametr	Naměřená hodnota
Průměrná rychlost	544,1 kb/s
Průměrná hodnota zpoždění	24,42 ms
Průměrná hodnota kolísání zpoždění	8,292 ms
Přenesených dat	100%

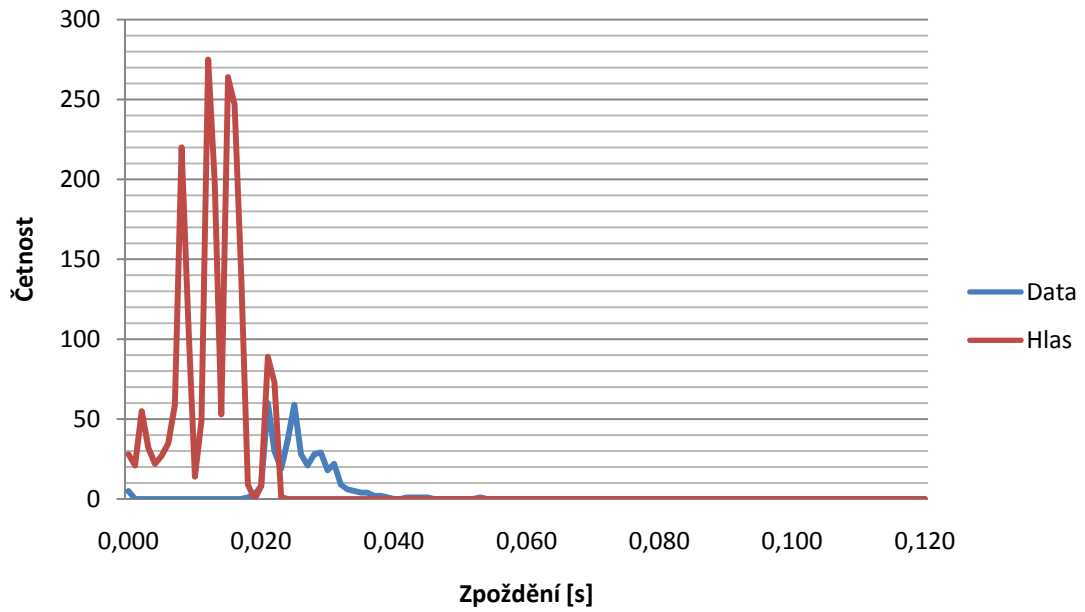
U měření s nižší zátěží se nevyskytly žádné problémy s přenosem ostatních dat. Jak hlasová tak i ostatní data byla přenesena se ztrátovostí 0%.

Z grafu znázorňujícího časový průběh hodnot zpoždění (obr. 5.8) je vidět, že ani maximální hodnota zpoždění nepřekračuje doporučenou hodnotu 150 ms u přenosu hlasových dat.



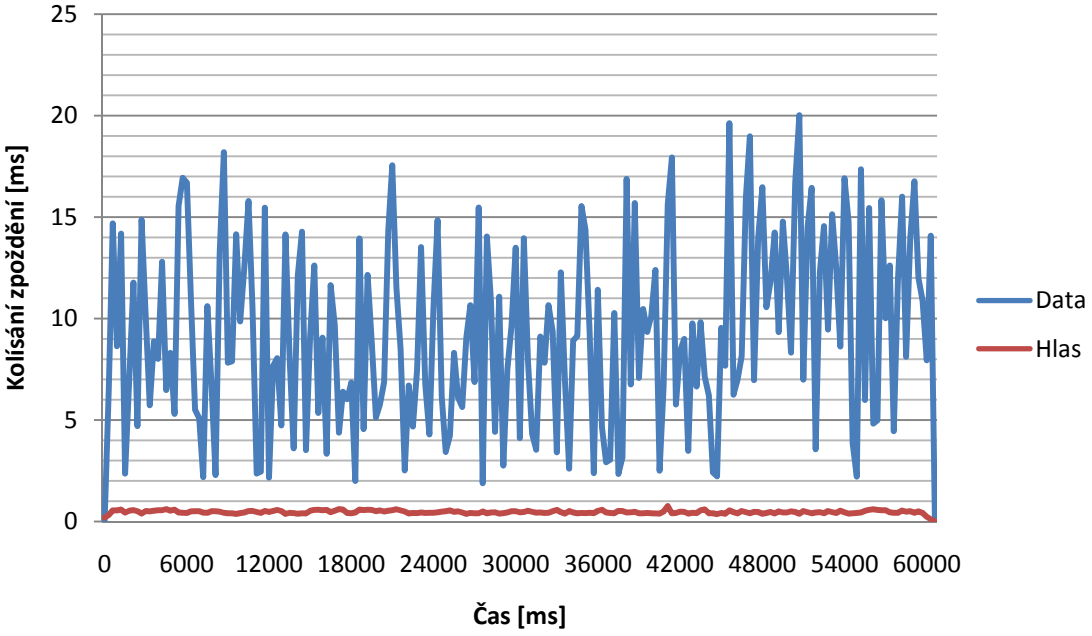
Obr. 5.8: Průběh zpoždění hlasových a ostatních dat

Také rozložení všech hodnot zpoždění, jak je vidět v grafu (obr. 5.9), leží v intervalu 0-150 ms, což značí podle ITU G.114 bezproblémový provoz hlasových služeb.



Obr. 5.9: Histogram zpoždění hlasových a ostatních dat

U měření s takovouto zátěží se nastavení mechanismu prioritních front příliš neprojeví. Ostatně je to také vidět v grafu kolísání zpoždění na obr. 5.10.



Obr. 5.10: Průběh kolísání zpoždění hlasových a ostatních dat

5.4.3 Měření s vyšší zátěží

U tohoto měření byl generován síťový provoz složený z 20 toků hlasových dat a 10 toků ostatních dat. Očekává se značné zhoršení přenosových charakteristik paketů ostatních dat s protokolem TCP. U hlasových dat by se měla projevit jejich prioritizace nižšími hodnotami zpoždění a kolísání zpoždění. Výsledky měření jsou uvedeny v tabulkách 5.8 a 5.9.

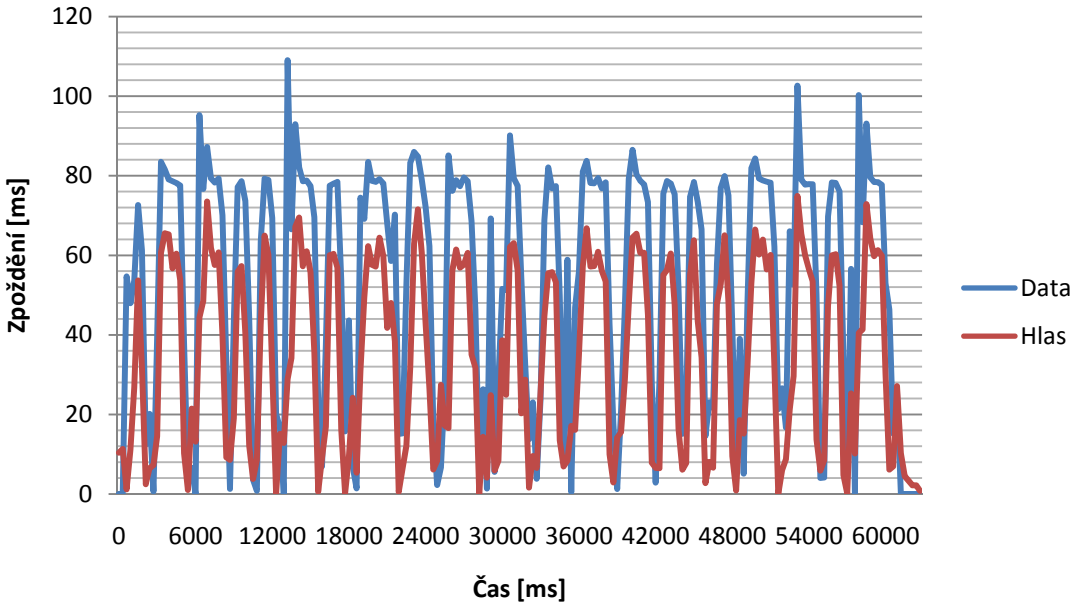
Tab. 5.8: Naměřené hodnoty sledovaných parametrů přenosu hlasových dat

Sledovaný parametr	Naměřená hodnota
Průměrná hodnota zpoždění	50,53 ms
Maximální hodnota zpoždění	280,9 ms
Průměrná hodnota kolísání zpoždění	1,059 ms
Ztrátovost paketů	0%

Tab. 5.9: Naměřené hodnoty sledovaných parametrů přenosu ostatních dat

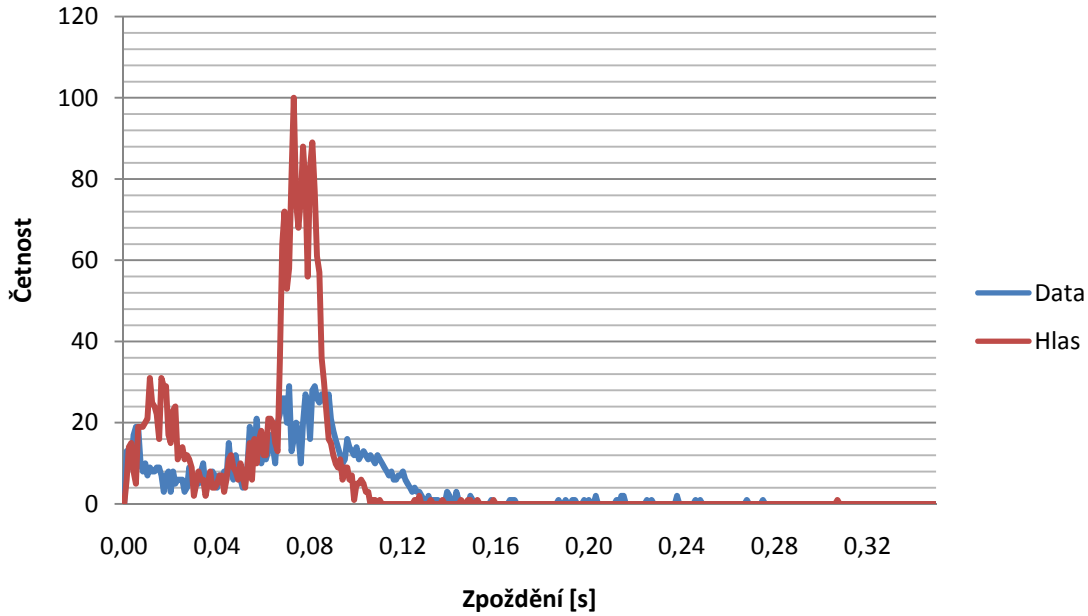
Sledovaný parametr	Naměřená hodnota
Průměrná rychlost	543,9 kb/s
Průměrná hodnota zpoždění	203,9 ms
Průměrná hodnota kolísání zpoždění	34,203 ms
Přenesených dat	100%

Z grafu na obr. 5.11 i výsledků naměřených hodnot v tabulkách 5.8 a 5.9 je patrný nárůst hodnot zpoždění u ostatních dat. Zde se již projevuje nastavení mechanismu PQ, tedy upřednostňování hlasových dat před ostatními. Oproti měření s mechanismem FIFO došlo ke snížení hodnot zpoždění u hlasových dat a naopak zvýšení průměrné hodnoty zpoždění u ostatních dat.



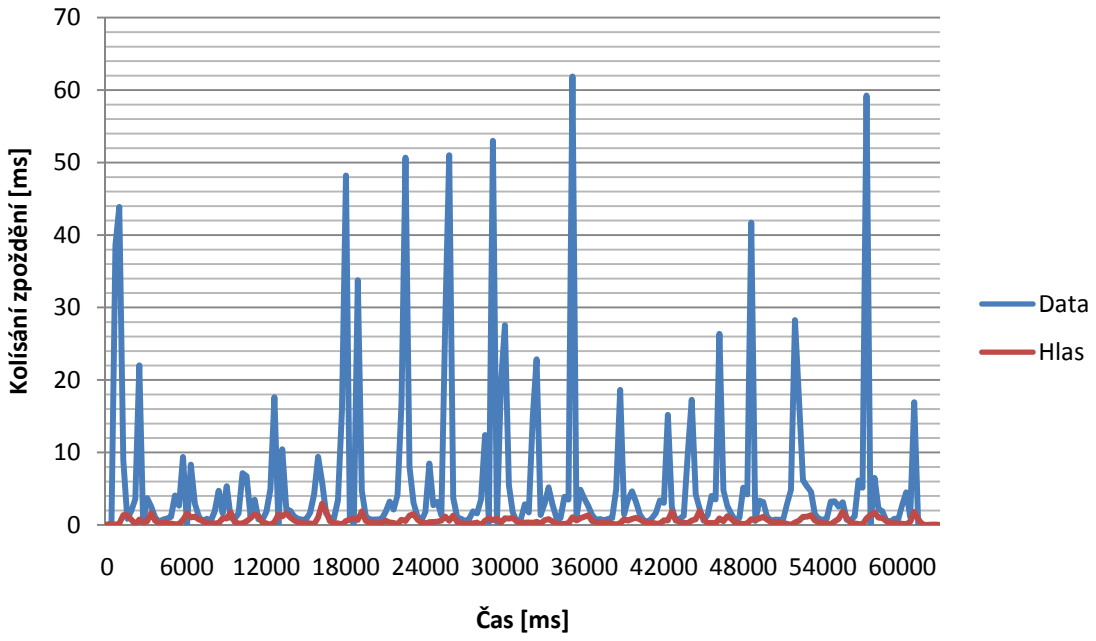
Obr. 5.11: Průběh zpoždění hlasových a ostatních dat

Graf na obr. 5.12 ukazuje četnost výskytu jednotlivých hodnot zpoždění. Přes 98% hodnot zpoždění je v intervalu 0-150 ms, zbývající 2% je v intervalu 150-400 ms. ITU G.114 sice považuje zpoždění 150-400 ms za možné, ale službu označuje jako degradovanou. V tomto případě by ale vzhledem k nízké četnosti měl být přenos hlasových dat bezproblémový.



Obr. 5.12: Histogram zpoždění hlasových a ostatních dat

Následující graf na obr. 5.13 ukazuje časový průběh hodnot kolísání zpoždění. Patrné je především téměř dvojnásobné zvýšení hodnot u ostatních dat oproti výsledkům měření s mechanismem FIFO. Hlasová data naopak vykazují mírně příznivější hodnotu. Jde o další důkaz funkčnosti mechanismu PQ při přenosu hlasových dat.



Obr. 5.13: Průběh kolísání zpoždění hlasových a ostatních dat

5.4.4 Zhodnocení

Měření dopadlo podle očekávání. Především u vyšší zátěže se projevila funkčnost mechanismu prioritních front. Priorita přiřazená hlasovým datům způsobila, že byla tato data odesílána přednostně před daty s nižší prioritou. Výsledkem jsou nižší hodnoty zpoždění a kolísání zpoždění u hlasových dat a naopak nárůst těchto hodnot u neupřednostňovaných ostatních dat.

5.5 Mechanismus CBWFQ

U předchozího mechanismu prioritních front je možné bez problémů a jednoduše prioritizovat provoz hlasových dat. Bohužel tato absolutní přednost prioritních dat před jinými daty může být i na škodu. Proto byl vytvořen mechanismus vážených front, který je schopen redukovat dopad přenosu objemných toků dat na přenosové charakteristiky jiných, menších toků.

5.5.1 Konfigurace

Protože je mechanismus CBFWQ class-based, musíme klasifikovat provoz do tříd. Nejdříve tedy vybereme provoz s UDP pakety:

```
R1(config)#access-list 101 permit udp any any
R1(config)#class-map match-all class-ef
R1(config-cmap)#match access-group 101
```

Dále musíme hlasový provoz označkovat. Vytvoříme politiku s názvem set-dscp a poté jí přiřadíme třídu s klasifikovaným provozem:

```
R1(config)#policy-map set-dscp
R1(config-pmap)#class class-ef
R1(config-pmap-c)set ip dscp ef
```

Následně přiřadíme nově označkovanému provozu třídu:

```
R1(config)#class-map match-all class-voice
R1(config-cmap)#match ip dscp ef
```

Poté už můžeme začít definovat mapu politik pro námi vybraný provoz. Hlasovým datům přiřadíme 45% a ostatním datům 30% z dostupné šířky pásma. Zároveň nastavíme maximální délky front. Pro hlasová data je délka 240 paketů, pro ostatní data 170:

```
R1(config)#policy-map qos-cbwfq
R1(config-pmap)#class class-voice
R1(config-pmap-c)#bandwidth percent 45
R1(config-pmap-c)#queue-limit 240
R1(config-pmap)#class class-default
R1(config-pmap-c)#bandwidth percent 30
R1(config-pmap-c)#queue-limit 170
```

Nakonec vytvořenou mapu politik aplikujeme na rozhraní:

```
R1(config-if)#service-policy output qos-cbwfq
```

Kontrola konfigurace aplikované na rozhraní:

```
FastEthernet0/0 is up, line protocol is up
  Hardware is Gt96k FE, address is 001b.d505.652c (bia 001b.d505.652c)
  Description: SMER R2
  Internet address is 10.1.2.1/30
  MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec,
     reliability 255/255, txload 28/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Full-duplex, 10Mb/s, 100BaseTX/FX
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:02, output 00:00:00, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: Class-based queueing
  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
     Conversations  0/3/256 (active/max active/max total)
     Reserved Conversations 2/2 (allocated/max allocated)
     Available Bandwidth 0 kilobits/sec
  5 minute input rate 31000 bits/sec, 60 packets/sec
  5 minute output rate 1123000 bits/sec, 383 packets/sec
     116843 packets input, 7046753 bytes
     Received 436 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 watchdog
     0 input packets with dribble condition detected
     838167 packets output, 295309502 bytes, 0 underruns
     0 output errors, 0 collisions, 6 interface resets
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
```

5.5.2 Měření s nižší zátěží

Při tomto měření nejsou očekávány nedostatky v kvalitě přenosu. Síť by měla být schopna bez problémů přenést celý generovaný tok s garancí nastavené QoS. Naměřené hodnoty jednotlivých parametrů přenosu jsou uvedeny v tab. 5.10 pro hlasová data a v tab. 5.11 pro ostatní data. Oproti měření s mechanismem PQ nedosahuje CBWFQ až tak dobré výsledky u zpoždění a kolísání zpoždění. To je zapříčiněno rozdílným přístupem k síťovému provozu. Dle tabulek 5.10 a 5.11 je vidět ztrátovost paketů 0% u obou typů datových toků.

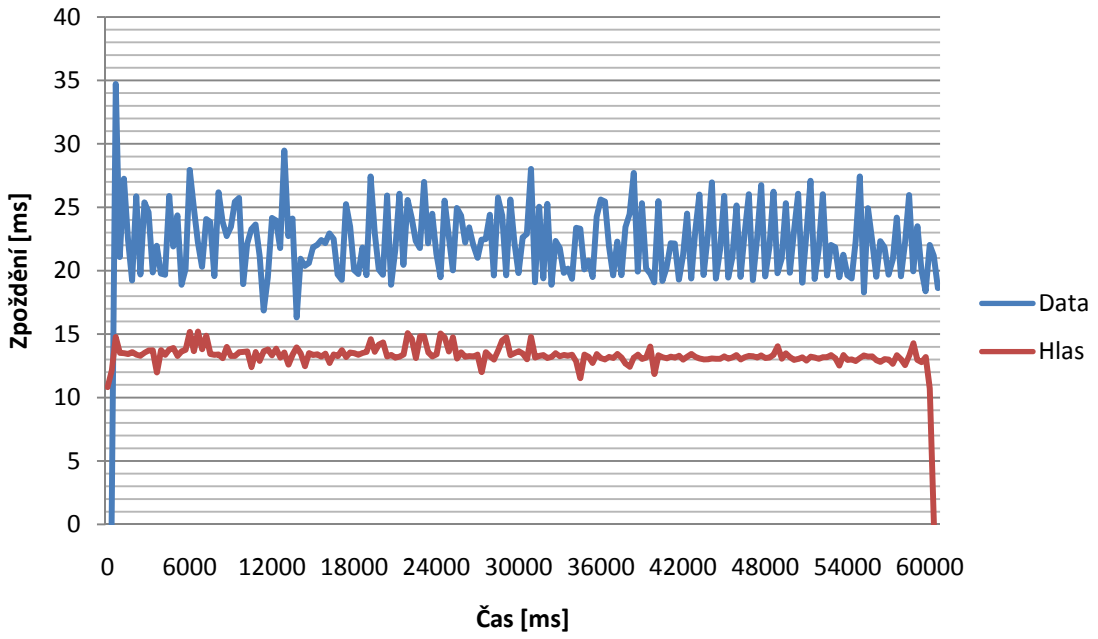
Tab. 5.10: Naměřené hodnoty sledovaných parametrů přenosu hlasových dat

Sledovaný parametr	Naměřená hodnota
Průměrná hodnota zpoždění	13,37 ms
Maximální hodnota zpoždění	87,00 ms
Průměrná hodnota kolísání zpoždění	0,442 ms
Ztrátovost paketů	0%

Tab. 5.11: Naměřené hodnoty sledovaných parametrů přenosu ostatních dat

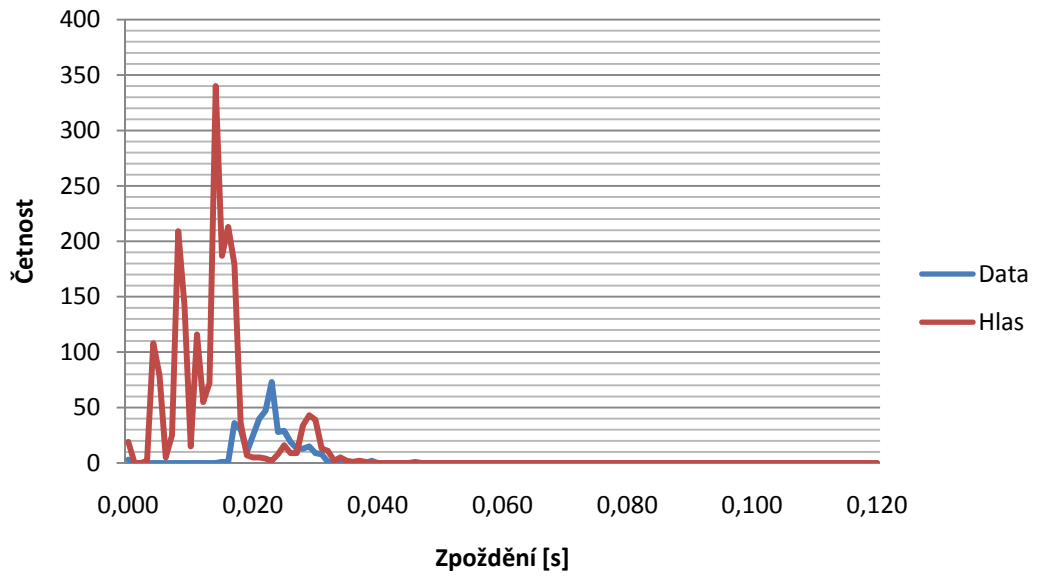
Sledovaný parametr	Naměřená hodnota
Průměrná rychlost	544,1 kb/s
Průměrná hodnota zpoždění	22,36 ms
Průměrná hodnota kolísání zpoždění	8,579 ms
Přenesených dat	100%

Hodnoty zpoždění u hlasových dat jsou mírně zvýšené a mají větší rozptyl, jak je vidět v grafu na obr. 5.14. Oproti tomu ostatní data vykazují mírně příznivější hodnoty zpoždění, než tomu bylo u mechanismu prioritních front.



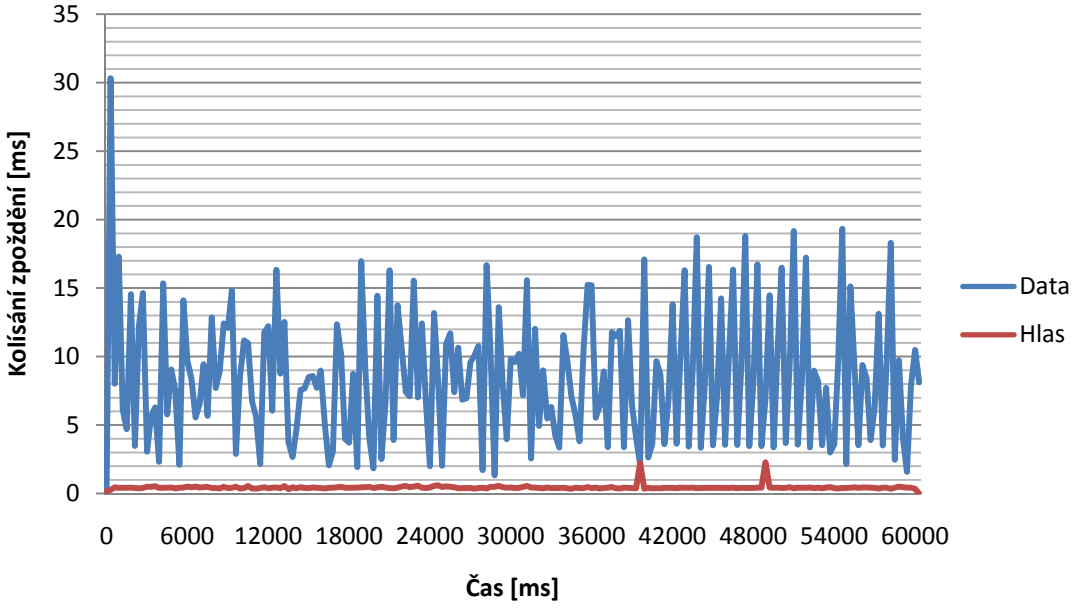
Obr. 5.14: Průběh zpoždění hlasových a ostatních dat

Četnost jednotlivých hodnot zpoždění (obr. 5.15) se opět jako u všech měření s nižší zátěží nachází v intervalu 0-150 ms, což značí zcela bezproblémový provoz hlasových služeb.



Obr. 5.15: Histogram zpoždění hlasových a ostatních dat

Graf na obr. 5.16 nabízí porovnání jednotlivých hodnot kolísání zpoždění mezi hlasovými a ostatními daty.



Obr. 5.16: Průběh kolísání zpoždění hlasových a ostatních dat

5.5.3 Měření s vyšší zátěží

V tomto měření se očekává vlivem nastavení šířky pásma ve prospěch hlasových dat zhoršení přenosových charakteristik ostatních dat. Jak je vidět z uvedených hodnot v tabulkách 5.12 a 5.13 a ze zobrazení časového průběhu zpoždění na obr. 5.17, původní

předpoklady výsledků měření se potvrdily. Větší objem přenášených dat se projevil na nárůstu hodnot zpoždění a především kolísání zpoždění u obou typů dat. U hlasových dat dochází také k 1,12% ztrátovosti paketů. Jde zřejmě o důsledek nastavení šířky pásma a délky front.

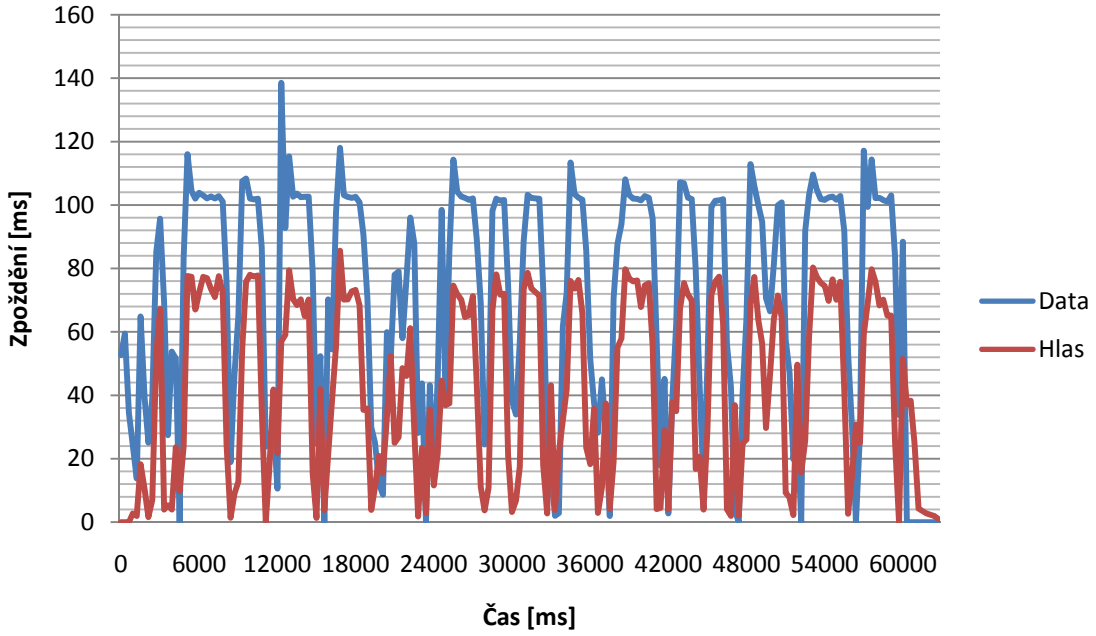
Tab. 5.12: Naměřené hodnoty sledovaných parametrů přenosu hlasových dat

Sledovaný parametr	Naměřená hodnota
Průměrná hodnota zpoždění	73,48 ms
Maximální hodnota zpoždění	250,6 ms
Průměrná hodnota kolísání zpoždění	1,007 ms
Ztrátovost paketů	1,12%

Tab. 5.13: Naměřené hodnoty sledovaných parametrů přenosu ostatních dat

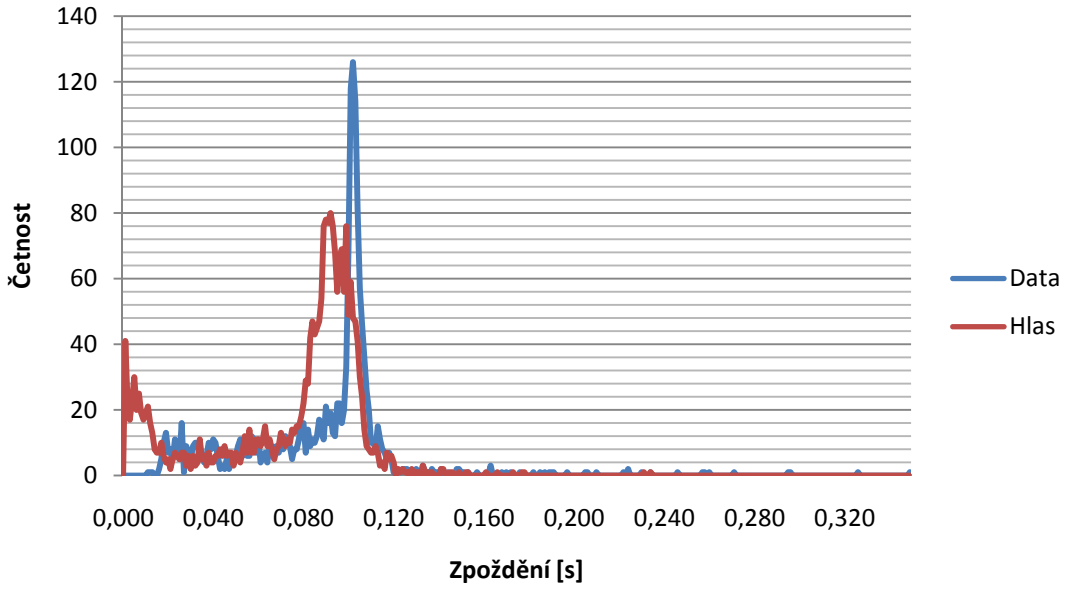
Sledovaný parametr	Naměřená hodnota
Průměrná rychlost	544,1 kb/s
Průměrná hodnota zpoždění	242,36 ms
Průměrná hodnota kolísání zpoždění	18,58 ms
Přenesených dat	100%

Časový průběh hodnot zpoždění hlasových a ostatních dat je uveden na obr. 5.17. Rozdíl mezi hodnotami zpoždění hlasových a ostatních dat je větší, než tomu bylo u mechanismu FIFO. Opět zde hraje roli přidělená šířka pásma jednotlivým typům provozu.



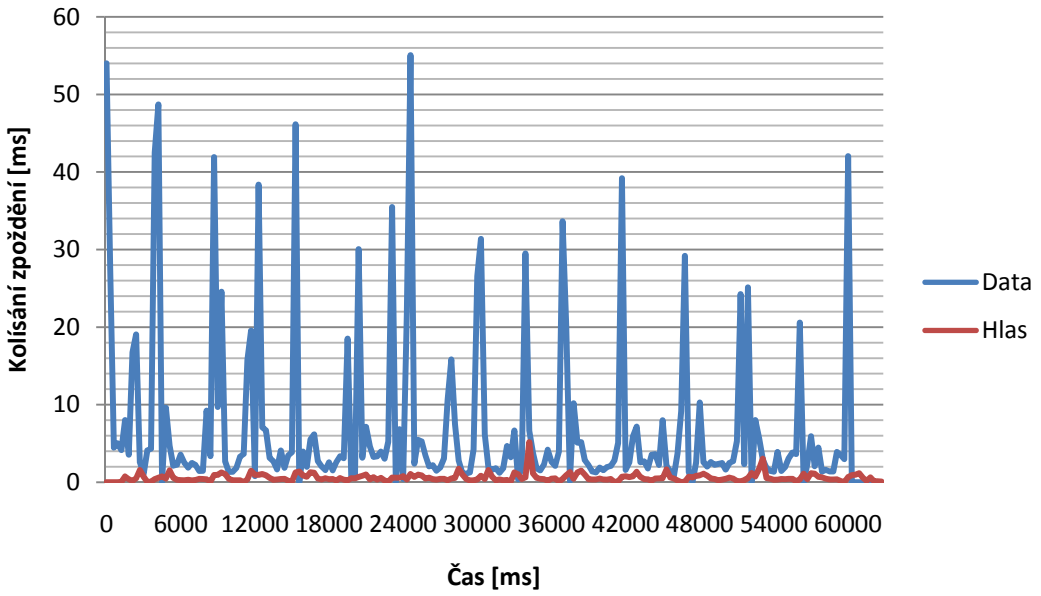
Obr. 5.17: Průběh zpoždění hlasových a ostatních dat

Graf 5.18 znázorňuje četnost jednotlivých hodnot zpoždění. Přes 92% hodnot zpoždění leží v intervalu 0-150 ms, 8% leží v intervalu 150-400 ms. Hlasový provoz by i při těchto 8% tzv. degradovaného toku by měl být přijatelný.



Obr. 5.18: Histogram zpoždění hlasových a ostatních dat

Jak je vidět z grafu na obr. 5.19, kolísání zpoždění jak u hlasových tak i u ostatních dat je výrazně nižší než u mechanismu PQ. Kolísání zpoždění u hlasových dat bez problémů splňuje podmínku ITU G.114, tedy velikost do 30 ms.



Obr. 5.19: Průběh kolísání zpoždění hlasových a ostatních dat

5.5.4 Zhodnocení

Vhodnou volbou přenosové šířky pásma lze pomocí CBWFQ upřednostňovat několik toků mezi sebou, bez toho aby agresivní prioritní tok pro sebe zabral veškerou přenosovou kapacitu jak je tomu u PQ. CBWFQ se tak stává univerzálnějším nástrojem v případě, že provoz tvoří několik typů datových toků (např. hlas a video přenášené v reálném čase).

5.6 Mechanismus LLQ

Mechanismus LLQ (Low Latency Queuing) v podstatě představuje další rozšíření možností mechanismu CBWFQ. K mechanismu vážených front přidává možnost striktně prioritizovat určitý datový tok na úrovni třídy. U této třídy provozu, na kterou je aplikován mechanismus LLQ, jsou garantovány nízké hodnoty zpoždění a kolísání zpoždění. Toky této třídy jsou v případě překročení vymezené šířky pásma zpracovány nástrojem policer. Na druhou stranu u této třídy s LLQ nemůžeme použít WRED ani definovat velikost fronty.

5.6.1 Konfigurace

LLQ je vychází z mechanismu CBFWQ. Proto se v základě konfigurace neliší od mechanismu vážených front. Nejprve musíme klasifikovat provoz, vybereme tedy provoz s UDP pakety:

```
R1(config)#access-list 101 permit udp any any
R1(config)#class-map match-all class-ef
R1(config-cmap)#match access-group 101
```

Dále musíme hlasový provoz označkovat. Vytvoříme politiku s názvem set-dscp a poté ji přiřadíme třídu s klasifikovaným provozem:

```
R1(config)#policy-map set-dscp
R1(config-pmap)#class class-ef
R1(config-pmap-c)set ip dscp ef
```

Následně přiřadíme nově označkovanému provozu třídu:

```
R1(config)#class-map match-all class-voice
R1(config-cmap)#match ip dscp ef
```

Nyní přistoupíme k definování mapy politik pro námi vybraný provoz. Nejprve tedy přiřadíme striktní prioritu hlasovým datům. Ostatním datům necháme přiřazenou šířku pásma 30% a délku fronty 170 paketů:

```
R1(config)#policy-map qos-llq
R1(config-pmap)#class class-voice
R1(config-pmap-c)#priority percent 45
R1(config-pmap)#class class-default
R1(config-pmap-c)#bandwidth percent 30
R1(config-pmap-c)#queue-limit 170
```

Nakonec vytvořenou mapu politik aplikujeme na rozhraní.

```
R1(config-if)#service-policy output qos-cbwfq
```

Kontrola nastavení mapy politik na výstupním rozhraní okrajového směrovače:

```
R1#show policy-map interface f0/0
 FastEthernet0/0
  Service-policy output: qos-llq
    Class-map: class-voice (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: ip dscp ef (46)
      Queueing
        Strict Priority
        Output Queue: Conversation 264
        Bandwidth 45 (%)
        Bandwidth 4500 (kbps) Burst 112500 (Bytes)
        (pkts matched/bytes matched) 0/0
        (total drops/bytes drops) 0/0
    Class-map: class-default (match-any)
      8 packets, 616 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: any
      Queueing
        Output Queue: Conversation 265
        Bandwidth 30 (%)
        Bandwidth 3000 (kbps)Max Threshold 170 (packets)
        (pkts matched/bytes matched) 0/0
        (depth/total drops/no-buffer drops) 0/0/0
```

5.6.2 Měření s nižší zátěží

Při měření stejně jako u všech předchozích s nízkou zátěží nejsou očekávány nějaké nedostatky v kvalitě přenosu. Síť by měla být schopna bez problémů přenést celý generovaný tok s garancí nastavené QoS. Tato měření slouží především pro účely porovnání s naměřenými hodnotami u vyšší zátěže. Výsledky jsou zpracovány v tab. 5.14 a 5.15.

Ani jeden z typů dat neměl problémy se ztrátovostí paketů. Hodnoty v podstatě odpovídají naměřeným hodnotám u mechanismu FIFO. Nastavení LLQ se u tak nízké zátěže nijak výrazně neprojevilo.

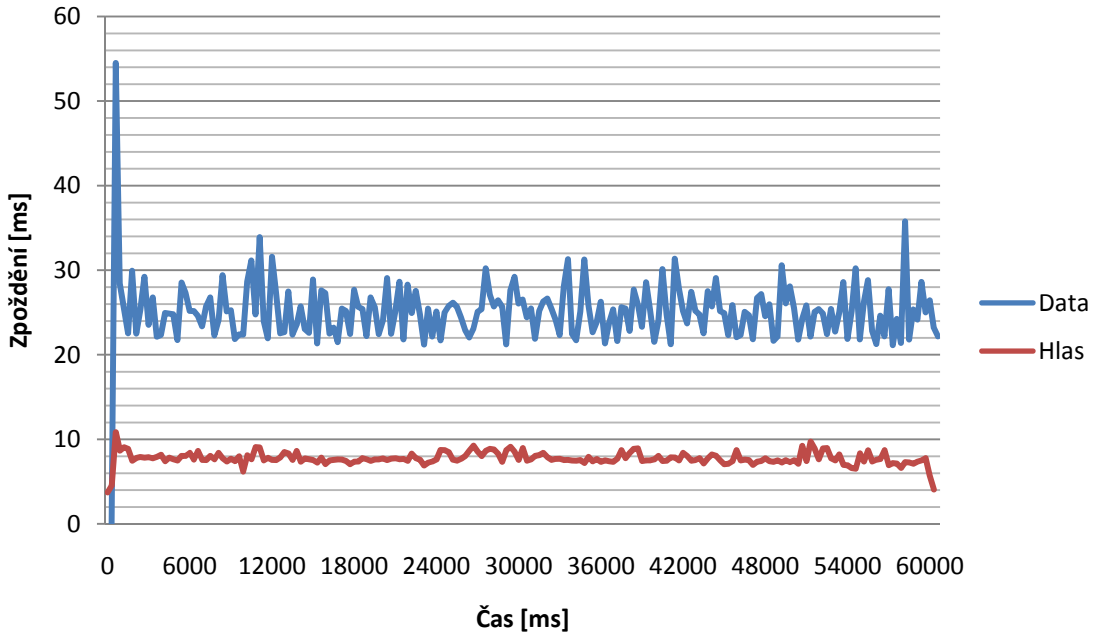
Tab. 5.14: Naměřené hodnoty sledovaných parametrů přenosu hlasových dat

Sledovaný parametr	Naměřená hodnota
Průměrná hodnota zpoždění	7,391 ms
Maximální hodnota zpoždění	48,10 ms
Průměrná hodnota kolísání zpoždění	0,457 ms
Ztrátovost paketů	0%

Tab. 5.15: Naměřené hodnoty sledovaných parametrů přenosu ostatních dat

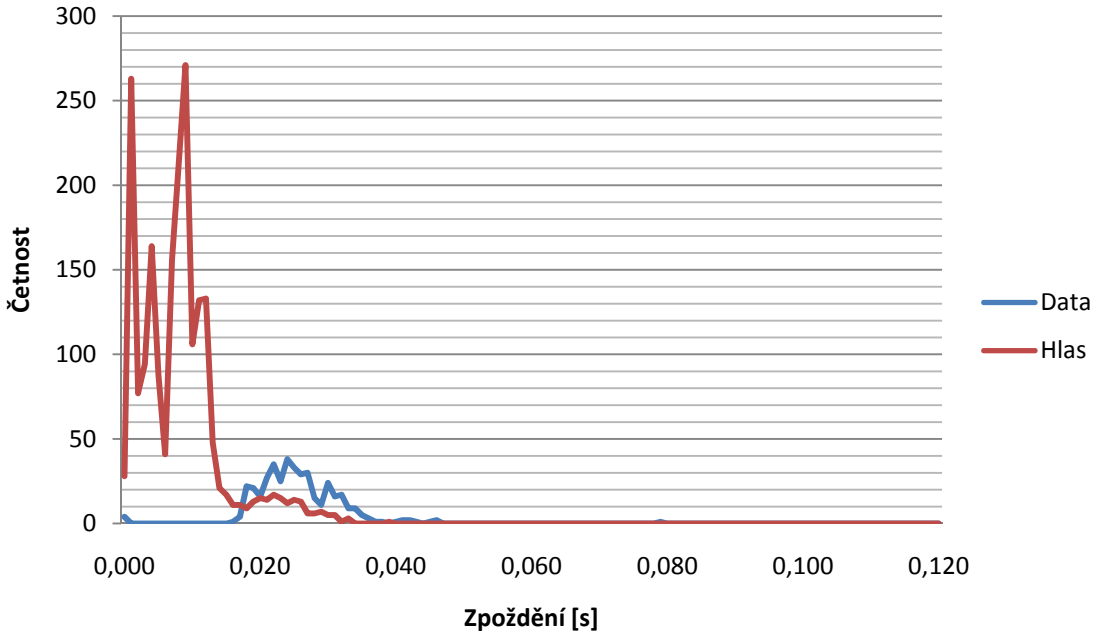
Sledovaný parametr	Naměřená hodnota
Průměrná rychlost	544,6 kb/s
Průměrná hodnota zpoždění	25,01 ms
Průměrná hodnota kolísání zpoždění	8,375 ms
Přenesených dat	100%

Graf na obr. 5.21 znázorňuje časový průběh hodnot zpoždění hlasových a ostatních dat. U hlasových dat tyto hodnoty nepřesahují interval 0-150 ms doporučovaný pro bezproblémový provoz hlasových služeb.



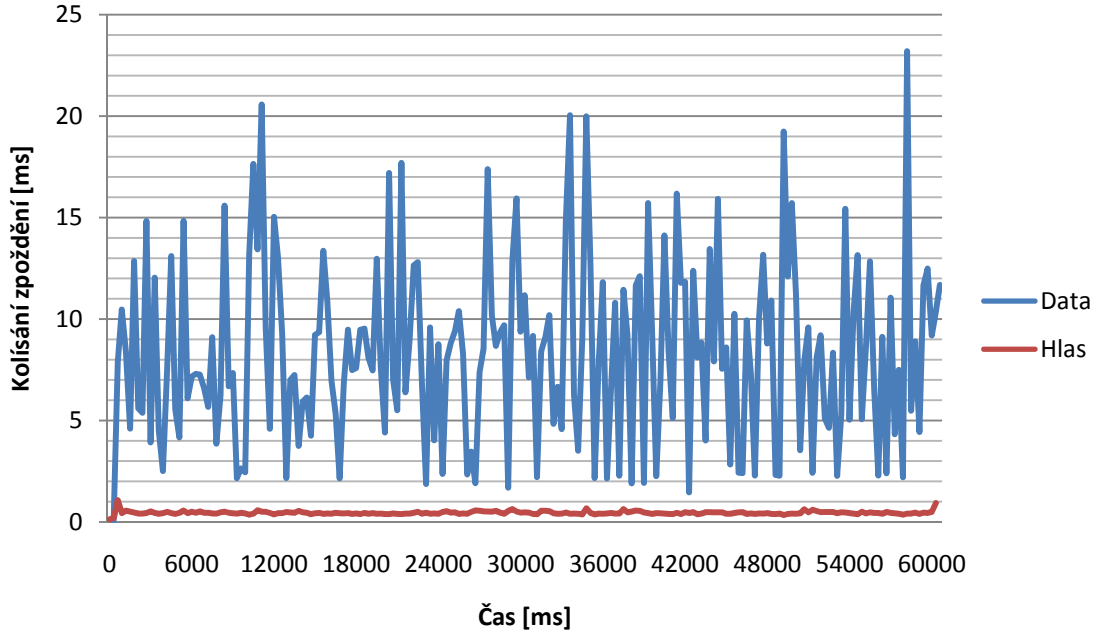
Obr. 5.21: Průběh zpoždění hlasových a ostatních dat

Četnost (100%) jednotlivých hodnot zpoždění (obr. 5.22) se opět jako u všech měření s nízkou zátěží objevuje v intervalu 0-150 ms.



Obr. 5.22: Četnost hodnot zpoždění hlasových a ostatních dat

Graf na obr. 5.23 nabízí porovnání jednotlivých hodnot kolísání zpoždění mezi hlasovými a ostatními daty



Obr. 5.23: Průběh kolísání zpoždění hlasových a ostatních dat

5.6.3 Měření s vyšší zátěží

V tomto měření se očekává vlivem nastavení šířky pásma ve prospěch hlasových dat mírné zhoršení přenosových charakteristik ostatních dat. Výsledné naměřené hodnoty jednotlivých sledovaných parametrů síťového provozu jsou uvedeny v tab. 5.16 a 5.17 podle typu dat. Za zmínku stojí příznivější hodnoty zpoždění a kolísání zpoždění u obou typů dat než tomu bylo u mechanismu CBWFQ.

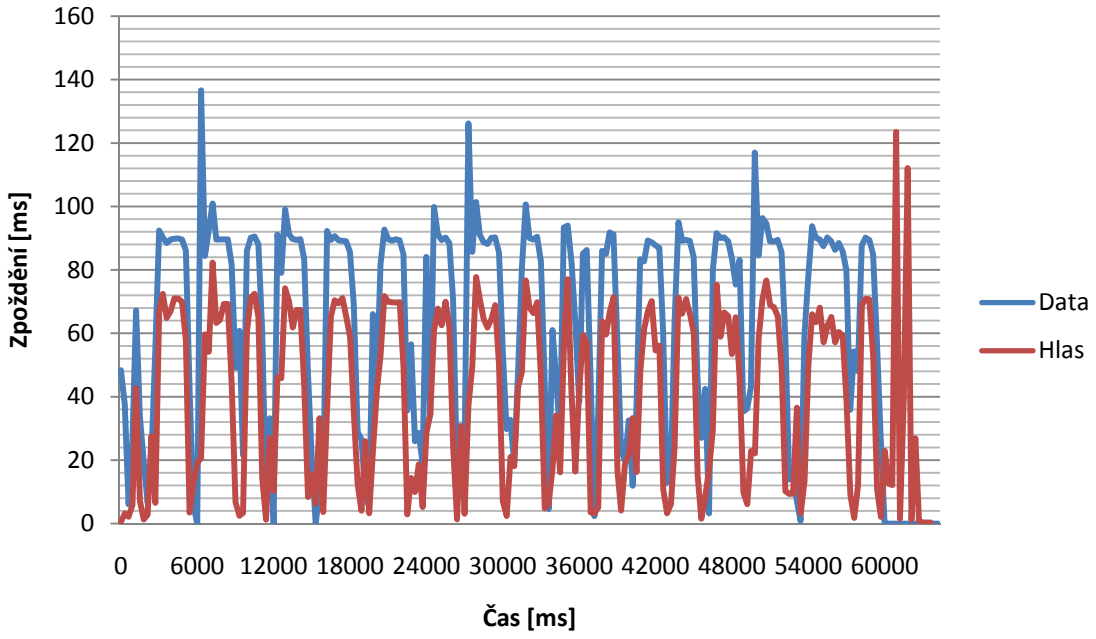
Tab. 5.16: Naměřené hodnoty sledovaných parametrů přenosu hlasových dat

Sledovaný parametr	Naměřená hodnota
Průměrná hodnota zpoždění	68,65 ms
Maximální hodnota zpoždění	124,3 ms
Průměrná hodnota kolísání zpoždění	0,905 ms
Ztrátovost paketů	0,02%

Tab. 5.17: Naměřené hodnoty sledovaných parametrů přenosu ostatních dat

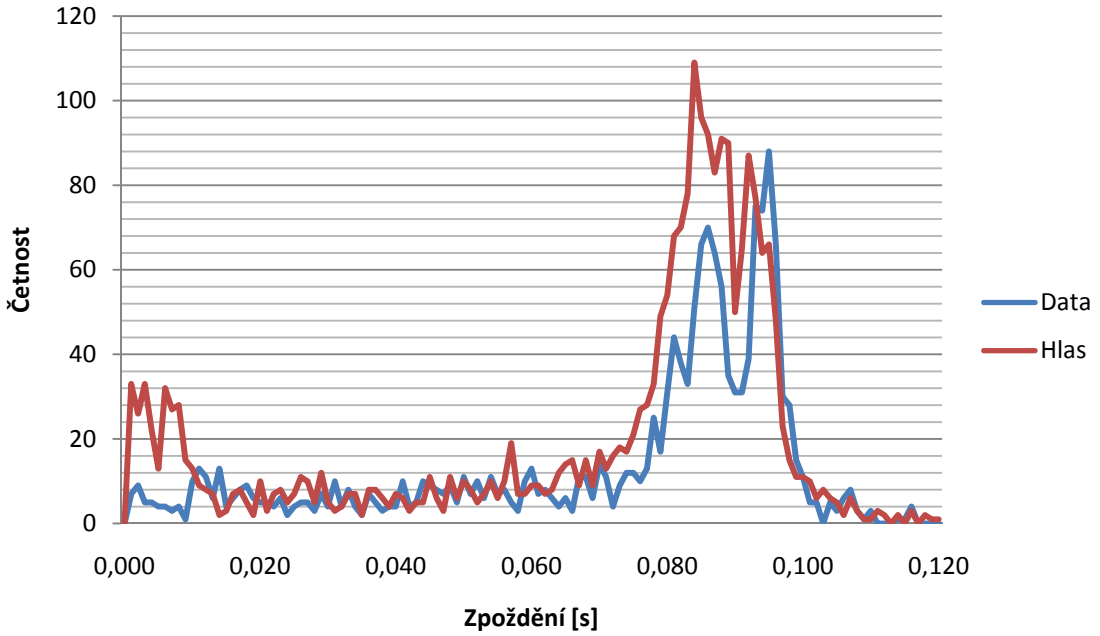
Sledovaný parametr	Naměřená hodnota
Průměrná rychlost	544,0 kb/s
Průměrná hodnota zpoždění	74,34 ms
Průměrná hodnota kolísání zpoždění	3,097 ms
Přenesených dat	100%

Časový průběh průměrných hodnot zpoždění pro 20 hlasových toků a 10 datových toků je znázorněn v grafu na obr. 5.24. U hlasových dat nepřesáhla hodnota zpoždění doporučovanou hranici 150 ms (ITU-G.114).



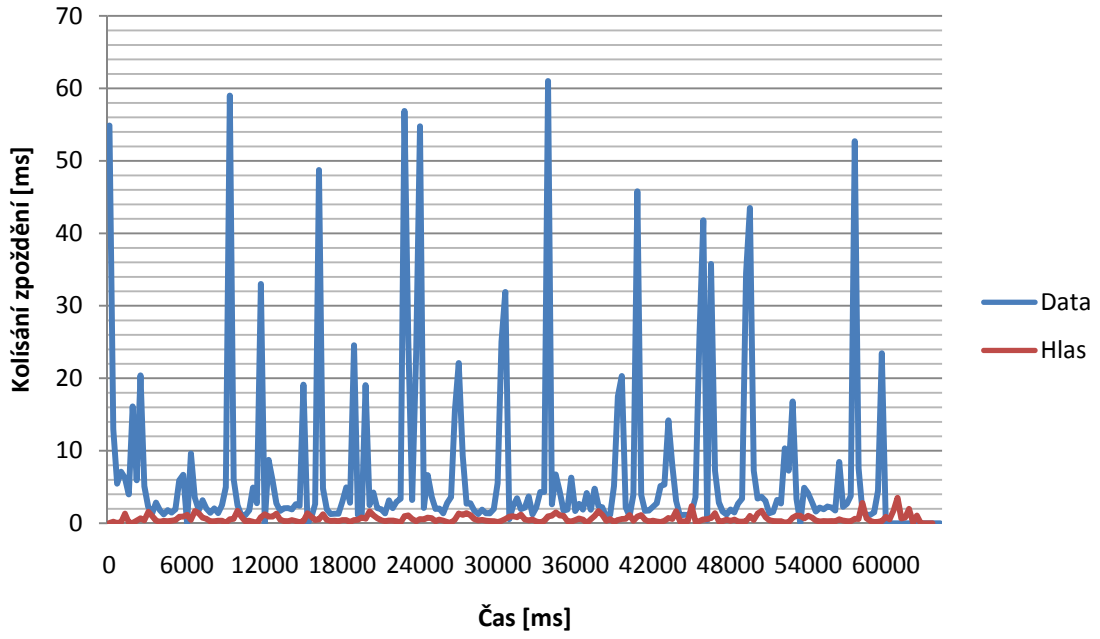
Obr. 5.24: Průběh zpoždění hlasových a ostatních dat

Histogram na obr. 5.25 znázorňuje rozložení jednotlivých hodnot zpoždění. Všechny hodnoty zpoždění u hlasových se nalézají v intervalu 0-150 ms.



Obr. 5.25: Četnost hodnot zpoždění hlasových a ostatních dat

Striktní priorita třídě se úspěšně projevila i u kolísání zpoždění hlasových dat. Naměřené hodnoty jsou skoro o desetinu nižší než u CBWFQ. Hodnoty kolísání zpoždění u hlasových dat nepřekročily doporučovanou hranici 30 ms.



Obr. 5.19: Průběh kolísání zpoždění hlasových a ostatních dat

5.6.4 Zhodnocení

Mechanismus LLQ je dnes z pohledu Cisca nejpoužívanějším nástrojem řízeného odesílání paketů. Spojuje v sobě výhody prioritní fronty a fronty s váženou obsluhou. Rovněž měření potvrdilo, že v uvedeném případě při daném generovaném provozu se jedná o nejoptimálnější řešení.

5.7 Volba mechanismu aktivní správy front

V rámci předchozích měření byly testovány mechanismy správy zahlcení. Kromě správy zahlcení nám ale implementace QoS u síťových zařízení Cisco umožňuje zahlcení předcházet. Nejčastěji uváděnými mechanismy jsou Drop-Tail, RED a WRED. Drop-Tail je standardně nastavený mechanismus, tudíž byl jeho vliv se projevil už v rámci testování implementace mechanismu FIFO. Pracuje tak, že v případě zahlcení zahazuje příchozí pakety tak dlouho, dokud zahlcení nepomine. Ke každému typu provozu se chová stejně [6]. Dnes se používají sofistikovanější metody založené na mechanismu RED (Random Early Detection). Cisco doporučuje využívat mechanismu RED pouze na linkách s rychlostí do 2 Mb/s. Pro větší přenosové rychlosti nabízí svoji implementaci mechanismu WRED (Weighted Random Early Detection), u kterého jsou pakety zahazovány podle profilů řídících se nastavenou váhou toku. Vzhledem k přenosovým rychlostem v testovací síti byl zvolen WRED. Původním plánem bylo vyzkoušet jej na nejoptimálnějším mechanismu řízení odesílání paketů. Protože se ale u LLQ nedá v prioritní třídě WRED použít, byl zvolen mechanismus CBWFQ.

5.8 Mechanismus WRED

Mechanismus WRED v sobě kombinuje možnosti algoritmu RED a prioritizace. Popis a princip mechanismu je součástí kapitoly 2.5.2. V praxi Cisco doporučuje nasazovat mechanismus na výstup rozhraní páteřních směrovačů DiffServ domény. Implementace Cisco obsahuje přednastavené profily pro IP Precedence a DSCP, nicméně je možné je upravovat. Vhodnou konfigurací rovněž umožňuje obejít vážené zahazování paketů, v podstatě návrat k holému mechanismu RED.

5.8.1 Konfigurace

Implementace WRED u směrovačů Cisco je tzv. class-based. Nejprve tedy musíme klasifikovat provoz a vybrat provoz s UDP pakety:

```
R1(config)#access-list 101 permit udp any any
R1(config)#class-map match-all class-ef
R1(config-cmap)#match access-group 101
```

Dále musíme hlasový provoz označkovat. Vytvoříme politiku s názvem set-dscp a poté jí přiřadíme třídu s klasifikovaným provozem:

```
R1(config)#policy-map set-dscp
R1(config-pmap)#class class-ef
R1(config-pmap-c)set ip dscp ef
```

Následně přiřadíme nově označkovanému provozu třídu:

```
R1(config)#class-map match-all class-voice
R1(config-cmap)#match ip dscp ef
```

Poté už můžeme začít definovat mapu politik pro námi vybraný provoz. Hlasovým datům přiřadíme 45% a ostatním datům 30% dostupné šířky pásma. Nastavíme pro oba provozy mechanismus WRED s ponechanými přednastavenými parametry:

```
R1(config)#policy-map qos-wred
R1(config-pmap)#class class-voice
R1(config-pmap-c)#bandwidth percent 45
R1(config-pmap-c)#random-detect dscp-based
R1(config-pmap)#class class-default
R1(config-pmap-c)#bandwidth percent 30
R1(config-pmap-c)#random-detect dscp-based
```

Nakonec vytvořenou mapu politik aplikujeme na rozhraní:

```
R1(config-if)#service-policy output qos-wred
```

Kontrola aplikování mapy politik na rozhraní páteřního směrovače:

```
FastEthernet0/0
  Service-policy output: qos-wred
    Class-map: class-voice (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: ip dscp ef (46)
      Queueing
        Output Queue: Conversation 265
        Bandwidth 45 (%)
        Bandwidth 4500 (kbps)
        (pkts matched/bytes matched) 0/0
        (depth/total drops/no-buffer drops) 0/0/0
         exponential weight: 9
         mean queue depth: 0

   dscp    Transmitted      Random drop      Tail drop    Minimum Maximum  Mark
           pkts/bytes       pkts/bytes       pkts/bytes    thresh  thresh  prob
     ef       0/0              0/0              0/0           36      40  1/10

    Class-map: class-default (match-any)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: any
      Queueing
        Output Queue: Conversation 266
        Bandwidth 30 (%)
        Bandwidth 3000 (kbps)
        (pkts matched/bytes matched) 0/0
        (depth/total drops/no-buffer drops) 0/0/0
         exponential weight: 9
         mean queue depth: 0

   dscp    Transmitted      Random drop      Tail drop    Minimum Maximum  Mark
           pkts/bytes       pkts/bytes       pkts/bytes    thresh  thresh  prob
default      0/0              0/0              0/0           20      40  1/10
```

5.8.2 Měření s nižší zátěží

Vzhledem ke generované zátěži by se mechanismus WRED neměl nijak výrazně projevit. Provoz by měl z pohledu přenosových charakteristik probíhat bezproblémově.

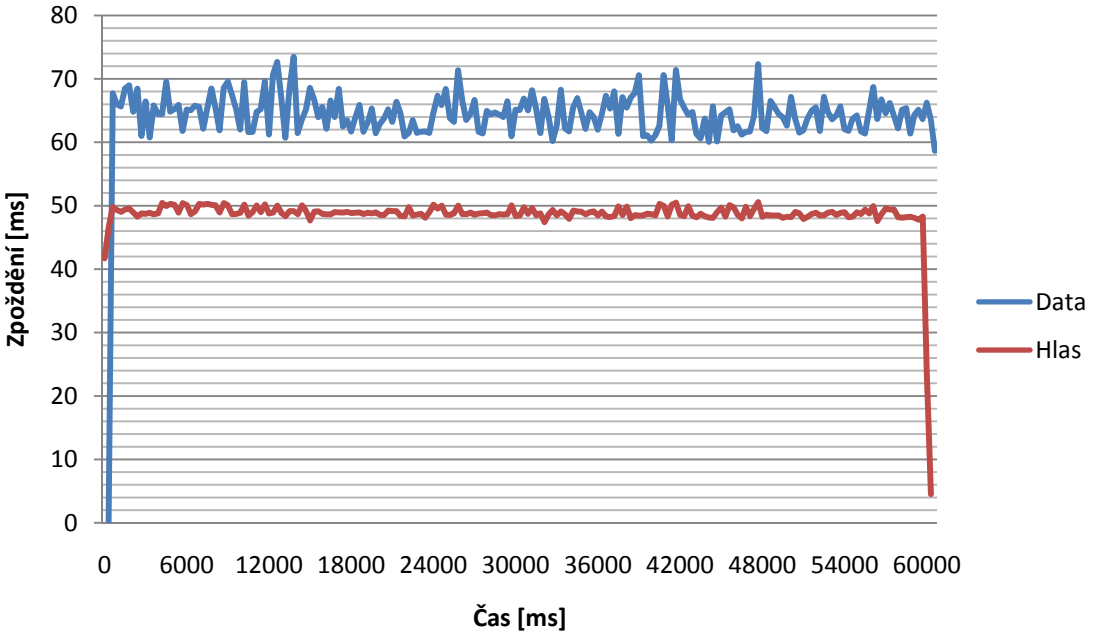
Jak je vidět z tabulek 5.18 a 5.19 a z grafu na obr. 5.26, je průměrná hodnota zpoždění mírně nižší než u měření přenosových charakteristik s implementací mechanismu CBWFQ bez WRED.

Tab. 5.18: Naměřené hodnoty sledovaných parametrů přenosu hlasových dat

Sledovaný parametr	Naměřená hodnota
Průměrná hodnota zpoždění	48,87 ms
Maximální hodnota zpoždění	70,7 ms
Průměrná hodnota kolísání zpoždění	0,453 ms
Ztrátovost paketů	0%

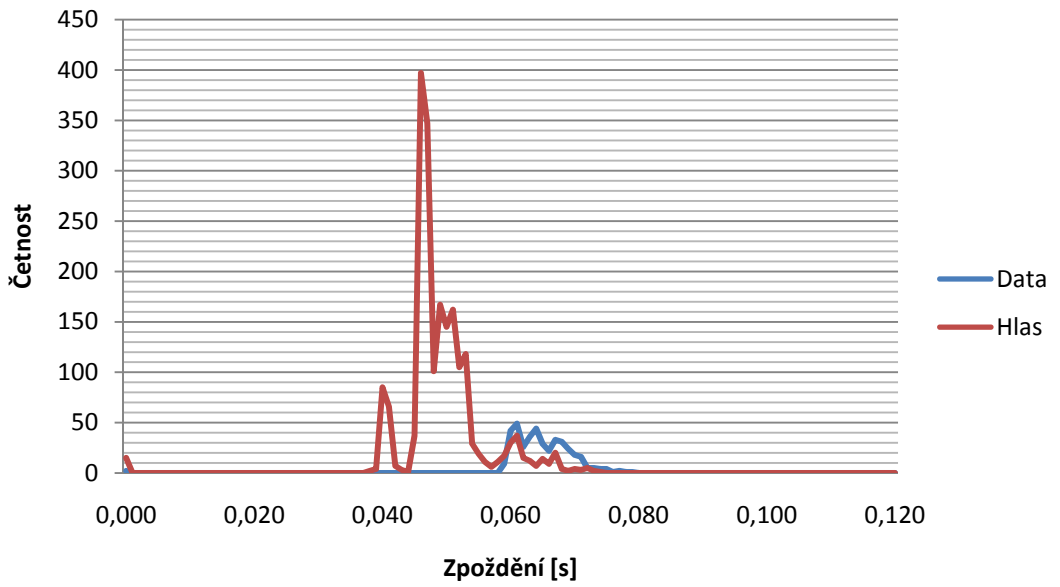
Tab. 5.19: Naměřené hodnoty sledovaných parametrů přenosu ostatních dat

Sledovaný parametr	Naměřená hodnota
Průměrná přenosová rychlost	544,6 kb/s
Průměrná hodnota zpoždění	64,55 ms
Průměrná hodnota kolísání zpoždění	8,801 ms
Přenesených dat	100%



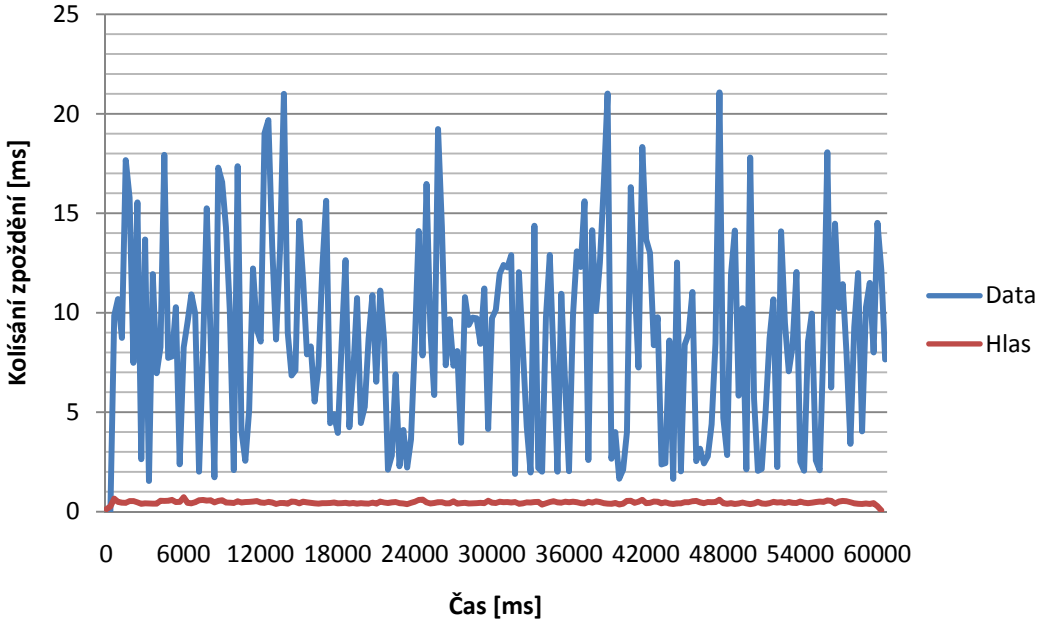
Obr. 5.26: Průběh zpoždění hlasových a ostatních dat

Hodnoty zpoždění u hlasu zdaleka nedosahují hranice 150 ms. Jak ukazuje histogram na obr. 5.27, je 100% hodnot v intervalu 0-150 ms. Hlasová data jsou v tomto případě doručena naprosto v pořádku.



Obr. 5.27: Histogram zpoždění hlasových a ostatních dat

Hodnoty zpoždění obou toků dat, jejichž časový průběh ukazuje graf na obr. 5.28, nevykazují žádné výrazné rozdíly mezi měřením s použitím WRED a bez jeho implementace.



Obr. 5.28: Průběh kolísání zpoždění hlasových a ostatních dat

5.8.3 Měření s vyšší zátěží

V tomto měření se očekává vlivem nastavení šířky pásma ve prospěch hlasových dat mírné zhoršení přenosových charakteristik ostatních dat.

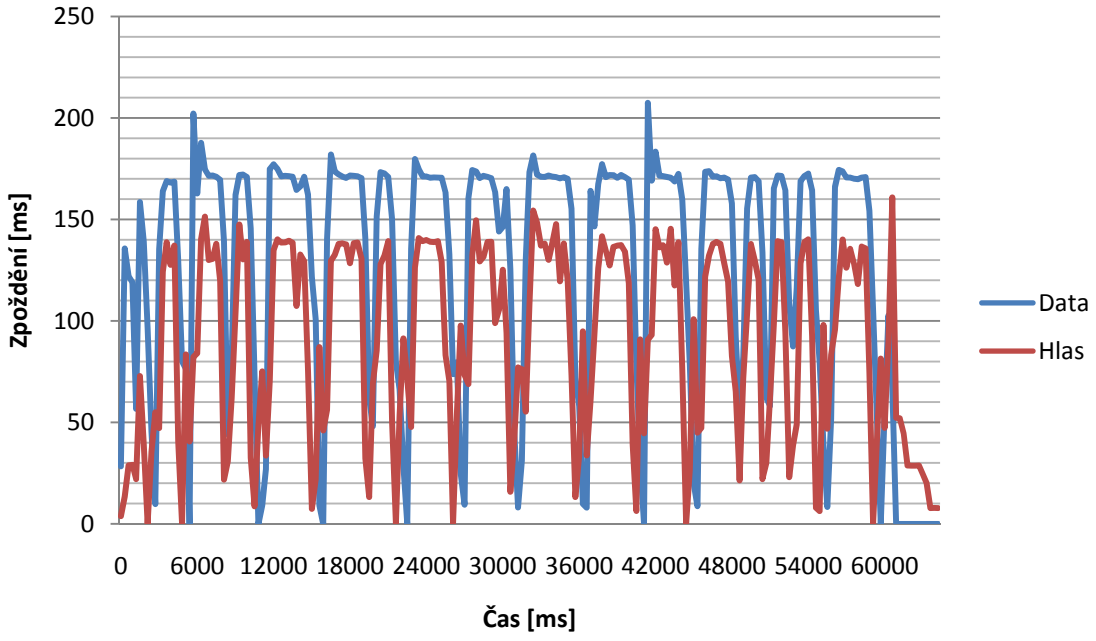
Z naměřených hodnot uvedených v tabulkách 5.20 a 5.21 je vidět, že hodnoty zpoždění jsou u obou typů dat zvýšené oproti předchozímu měření bez použití WRED. Časový průběh hodnot zpoždění je znázorněn v grafu na obr. 5.29.

Tab. 5.20: Naměřené hodnoty sledovaných parametrů přenosu hlasových dat

Sledovaný parametr	Naměřená hodnota
Průměrná hodnota zpoždění	152,1ms
Maximální hodnota zpoždění	405,2ms
Průměrná hodnota kolísání zpoždění	1,059ms
Ztrátovost paketů	0,01%

Tab. 5.21: Naměřené hodnoty sledovaných parametrů přenosu ostatních dat

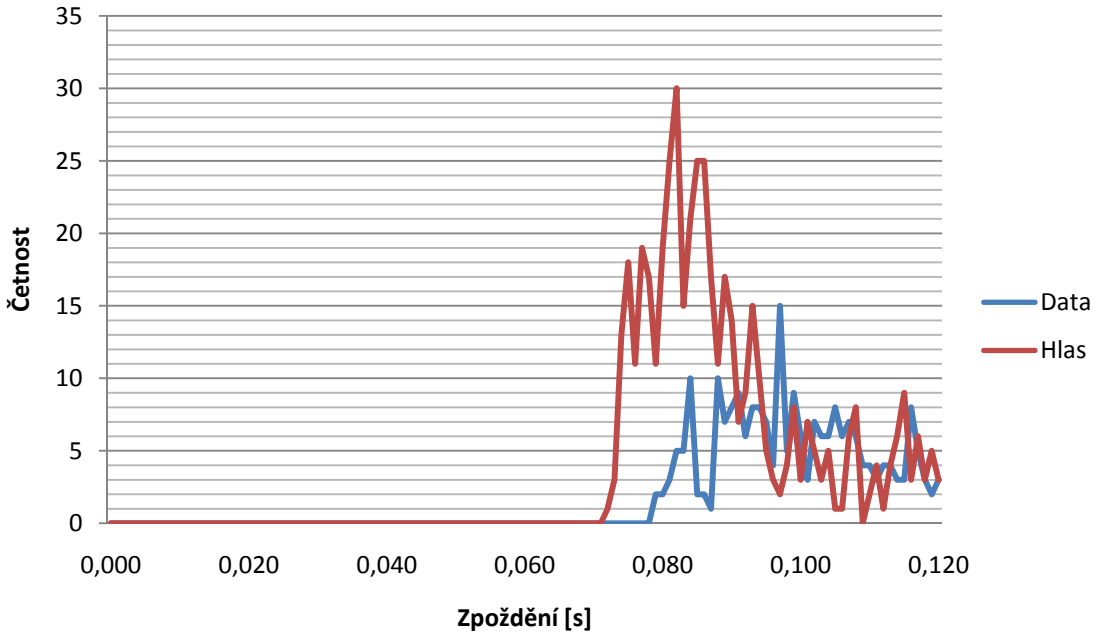
Sledovaný parametr	Naměřená hodnota
Průměrná přenosová rychlost	539,9kb/s
Průměrná hodnota zpoždění	158,8ms
Průměrná hodnota kolísání zpoždění	2,891 ms
Přenesených dat	97%



Obr. 5.29: Průběh zpoždění hlasových a ostatních dat

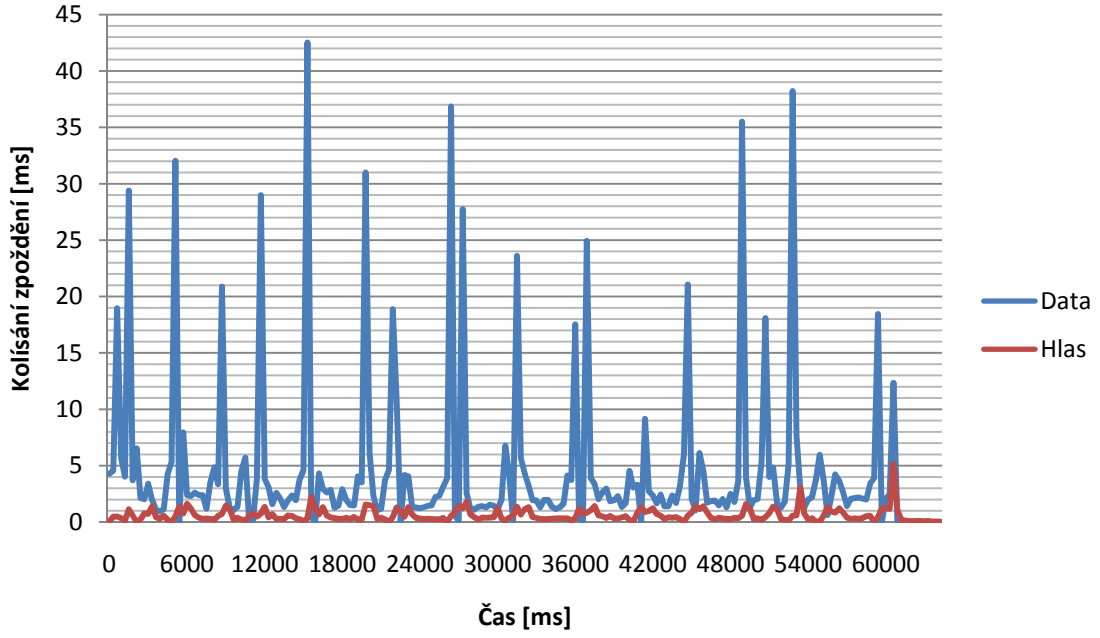
Histogram na obr. 5.30 znázorňuje četnost jednotlivých hodnot zpoždění. Přes 80% hodnot leží v intervalu 0-150 ms. Zbytek hodnot náleží intervalu 150-400 ms, který

označuje tyto hodnoty jako zpoždění degradované služby. V tomto případě zřejmě už bude poznat určité zhoršení kvality přenosu hlasových dat.



Obr. 5.30: Histogram zpoždění hlasových a ostatních dat

Graf na obr. 5.31 zobrazuje časový průběh kolísání zpoždění jednotlivých toků. V tomto případě jsou hodnoty kolísání zpoždění u ostatních dat mnohem nižší oproti měření bez implementace mechanismu WRED.



Obr. 5.31: Průběh kolísání zpoždění hlasových a ostatních dat

5.8.4 Zhodnocení

Mechanismus WRED pracuje podle váhy jednotlivých přednastavených profilů, v tomto případě DSCP profilu. V případě zahlcení začne podle této váhy zahazovat pakety. Především u měření s vyšší zátěží docházelo k zahazování paketů jako obraně před zahlcením.

5.9 Vyhodnocení provedených měření

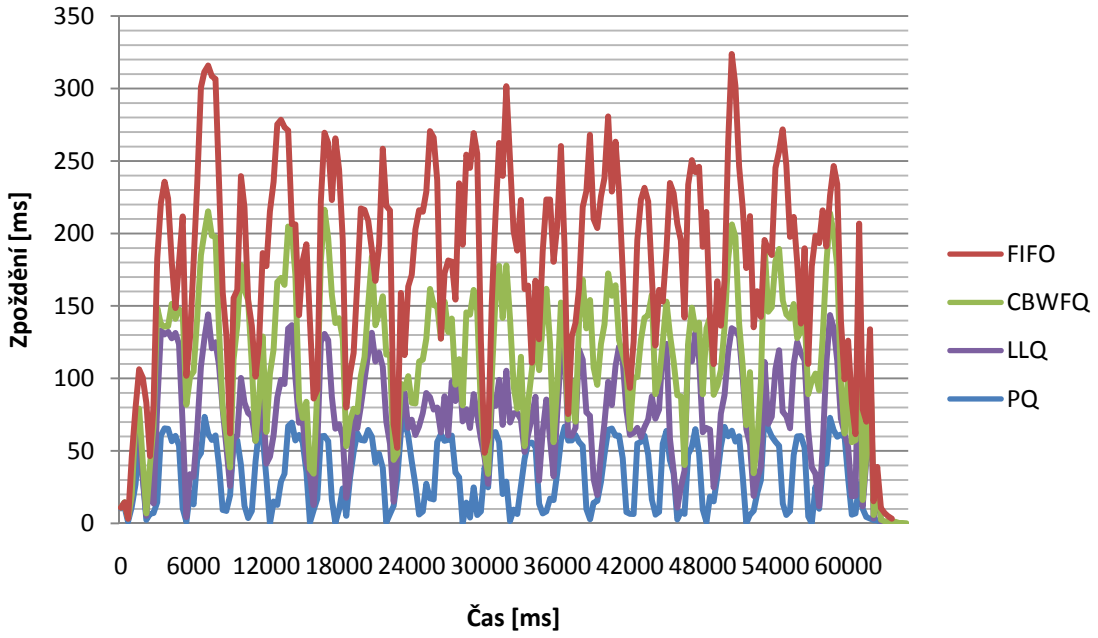
První měření probíhalo v rámci implementace mechanismu FIFO. Protože je mechanismus na FastEthernetových rozhraních standardně nastaven, šlo v podstatě o měření Best-Effort, tedy bez jakéhokoliv nastavování QoS. Veškerá manipulace mechanismu FIFO s pakety je určena pouze pořadím, ve kterém pakety přicházejí do fronty. V tom samém pořadí je také mechanismus zpracuje a pošle dál, jakmile se výstupní linka uvolní. V případě, že se fronta naplní dříve, než dojde k uvolnění linky, začne mechanismus pakety zahazovat. Protože mechanismus FIFO nedělá rozdíly mezi pakety s vysokou prioritou a pakety s nízkou prioritou, je zřejmé, že tento mechanismus není příliš vhodný pro přenos prioritizovaného provozu. Náhle špičky v provozu způsobují nejen velké zpoždění při doručování dat, ale dokonce i zahazování paketů bez ohledu na důležitost (prioritu) přenášených dat. I když mechanismus FIFO představoval první krok k větší kontrole nad síťovým provozem a nevyžaduje žádné zvláštní nastavování a konfiguraci, dnešním mnohem sofistikovanějším konvergovaným sítím už zdaleka nedostačuje.

Druhým implementovaným mechanismem byl PQ. Mechanismus prioritizuje jednotlivé typy paketů. Pakety jsou řazeny do čtyř front: high, medium, normal a low. Nejprve je zpracována fronta s nejvyšší prioritou, po jejím vyprázdnění se začne zpracovávat fronta s nižší prioritou. Pakety s vyšší prioritou jsou tak odesílány před pakety s nižší prioritou. Mechanismus je ideální v situacích, kdy potřebujeme přidělit určitému provozu absolutní prioritu.

Třetím testovaným mechanismem řazení do front je CBWFQ. Jeho výhodou je odstranění největšího nedostatku mechanismu prioritních front. Nedochází tak k situacím, kdy při velkém provozu dat s vysokou prioritou data s nižší prioritou „zamrznou“ a zůstanou nezpracovány. CBWFQ přiděluje jednotlivým datovým tokům konkrétní šířku přenosového pásma. Je možné tak upřednostňovat více typů provozů.

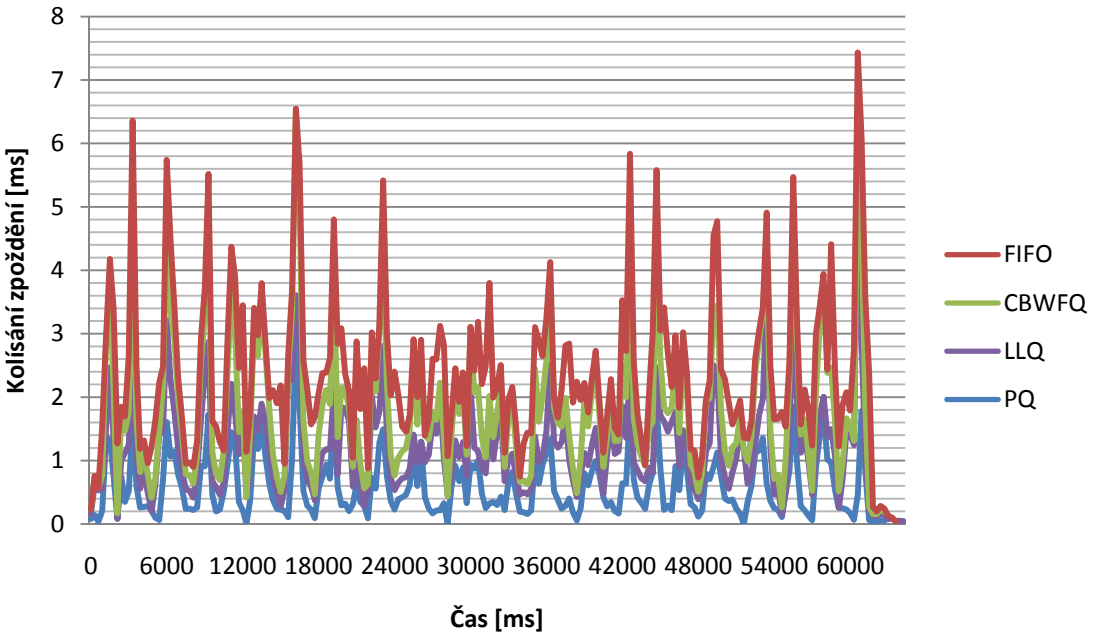
Posledním testovaným mechanismem je LLQ. Ten jde ještě dál, spojuje v sobě výhody jak CBWFQ tak PQ. Díky těmto dvěma přístupům k datům je ideální v případě síťového provozu s hlasovými daty.

Teoretický předpoklad před měřením řadil mechanismy front takto (od nejúčinnější po nejhorší): LLQ, CBWFQ, PQ, FIFO. Měření však tento předpoklad nepotvrdilo. Časové průběhy hodnot zpoždění všech testovaných mechanismů při přenosu hlasových dat při vyšší zátěži jsou znázorněny v grafu na obr. 5.32. Nejhorší výsledky hodnot zpoždění při vyšší generované zátěži byly naměřeny při implementaci mechanismu FIFO, tedy bez jakékoliv implementace nástrojů QoS. Nejlepší výsledky byly naměřeny při nasazení mechanismu prioritní fronty. Což bylo trochu překvapivé s ohledem na teoretický předpoklad.



Obr. 5.32: Časové průběhy hodnot zpoždění hlasových dat testovaných mechanismů

Dalším důležitým parametrem přenosu je kolísání zpoždění. Naměřené hodnoty kolísání zpoždění jednotlivých mechanismů při vyšší zátěži znázorňuje graf na obr. 5.33. Nejlepších hodnot opět dosáhl mechanismus prioritních front.



Obr. 5.33: Časové průběhy hodnot kolísání zpoždění přenosu hlasových dat testovaných mechanismů

Kromě mechanismů řízeného odesílání paketů proběhlo i otestování mechanismu předcházení zahlcení, neboli aktivní správy front, WRED. Měření potvrdilo funkčnost mechanismu, kdy při hrozícím zahlcení začal se zahazováním paketů ostatních dat a teprve poté s prioritizovaným tokem hlasových dat. Cisco nabízí několik dalších variant mechanismů aktivní správy front založených na algoritmu WRED. Liší se víceméně způsobem výběru prioritizovaného provozu, ať už je rozlišovacím znakem DSCP, IP priorita, typ datového toku aj.

6. ZÁVĚR

Cílem této práce bylo seznámení s možnostmi konfigurace mechanismů pro zajištění kvality služeb na síťových prvcích firmy Cisco. První kapitola této práce poskytuje přehled současného stavu problematiky zajištění kvality služeb v IP sítích. Následující kapitola se již podrobněji věnuje základnímu popisu jednotlivých charakteristik síťového provozu z pohledu QoS. Dále jsou obecně popsány jednotlivé nástroje technologie DiffServ pro zajištění kvality služeb v síti. Třetí kapitola se věnuje konfiguračním možnostem zajištění kvality služeb na síťových prvcích Cisco. Nastíněn je princip práce s modulárním řádkovým rozhraním i konfigurace tříd a map politik. V další kapitole je popsána zvolená metodika měření a popis scénářů měření. V páté kapitole je navržena topologie testovací sítě s popisem jednotlivých prvků. Následně jsou změřeny vybrané mechanismy řazení do front a ochrany proti zahlcení a jejich vliv na síťový provoz. Podrobně jsou popsány jednotlivé možnosti konfigurace a ty nejzajímavější zdokumentovány v tabulkách a grafech. Na základě provedených měření byl z testovaných mechanismů vybrán jako nejoptimálnější, mechanismus LLQ.

Vývoj jednotlivých nástrojů řízení odesílání paketů i aktivní správy front jde ruku v ruce s vývojem výkonnějších síťových prvků. Stejně jako byl WFQ nahrazen CBWFQ, PQ mechanismem LLQ, jsou už dnes pro změnu nahrazovány CBWFQ i LLQ novým mechanismem MDRR (Modified Deficit Round Robin), podporovaným zatím jen směrovači Cisco řady 12000. Stejná situace je i v případě WRED. Cisco nabízí dvě další varianty, DWRED (Distributed Weighted Random Early Detection) podporovanou na směrovačích Cisco řady 7000 a Flow-Based WRED. Jak je vidět, otestované mechanismy v této práci představují jen jakýsi základní přehled problematiky s ohledem na nasazení v konkrétních podmínkách síťového provozu a na konkrétních zařízeních.

SEZNAM LITERATURY

[1] ALVAREZ, Santiago. *QoS for IP/MPLS Networks*. First Printing. Indianapolis: Cisco Systems, Inc, 2006. 336 s. ISBN 1-58705-233-4.

[2] BLAKE, Steven, et al. *RFC 2475 : An Architecture for Defferentiated Services* [online]. IETF, 1998 [cit. 2009-11-10]. Text v angličtině.
URL: <http://tools.ietf.org/pdf/rfc2475.pdf>.

[3] BOUŠKA, Petr. *Cisco QoS 1 - úvod do Quality of Service a DiffServ* [online]. 2009 [cit. 2009-11-11]. Text v češtině. URL: <http://www.samuraj-cz.com/clanek/cisco-qos-1-uvod-do-quality-of-service-a-diffserv/>.

[4] BOUŠKA, Petr. *Cisco QoS 2 – Classification and Marking, Modular QoS CLI* [online]. 2009 [cit. 2009-11-11]. Text v češtině. URL: <http://www.samuraj-cz.com/clanek/cisco-qos-2-classification-and-marking-modular-qos-cli/>.

[5] BOUŠKA, Petr. *Cisco QoS 3 – omezování rychlosti – Policing, Shaping* [online]. 2009 [cit. 2009-11-11]. Text v češtině. URL: <http://www.samuraj-cz.com/clanek/cisco-qos-3-omezovani-rychlosti-policing-shaping/>.

[6] BOUŠKA, Petr. *Cisco QoS 4 – garance rychlostí, řazení do front* [online]. 2009 [cit. 2009-11-11]. Text v češtině. URL: <http://www.samuraj-cz.com/clanek/cisco-qos-4-garance-rychlosti-razeni-do-front-queuing/>.

[7] BOUŠKA, Petr. *Cisco QoS 5 –QoS na switchi, MLS, SRR, Auto QoS* [online]. 2009 [cit. 2009-11-11]. Text v češtině. URL: <http://www.samuraj-cz.com/clanek/cisco-qos-5-qos-na-switchi-mls-srr-auto-qos/>.

[8] BOUŠKA, Petr. *Cisco QoS 6 –praktické příklady použití QoSu* [online]. 2009 [cit. 2009-11-11]. Text v češtině. URL: <http://www.samuraj-cz.com/clanek/cisco-qos-6-prakticke-priklady-pouziti-qosu/>.

[9] BOUŠKA, Petr. *Cisco QoS 7 –doplňující informace* [online]. 2009 [cit. 2009-11-11]. Text v češtině. URL: <http://www.samuraj-cz.com/clanek/cisco-qos-7-doplujici-informace/>.

[10] BROWN, Ian; DOOLEY, Kevin. *Cisco Cookbook*. Sebastopol: O'Reilly & Associates, Inc., 2003. 912 s. ISBN 0-596-00367-6.

[11] CASTELLI, Matthew. *LAN Switching: First-step*. Indianapolis: Cisco Systems, Inc, 2004. 408 s. ISBN 1-58720-100-3.

[12] CESNET. *QoS v IP* [online]. 2001. Praha : CESNET, [2001] [cit. 2009-11-11]. Text v češtině. URL: <http://www.cesnet.cz/doc/zprava2001/qosip.html>.

[13] *Cisco IOS Quality of Service Solutions Configuration Guide: Release 12.4*. [s.l.]: Cisco Systems, Inc, 2009. 1014 s.

[14] ČENŠČÁK, Pavol. *Architektúry Quality of Service v IP sieťach. : QoS nástroje v sieťových prvkoch Cisco*. Žilina : UNIZA UIKT, 2006. 66 s. Žilinská univerzita v žiline. Vedoucí bakalářské práce Ing. Pavel Segeč, PhD.

[15] HOŠEK, Jiří. *Zajištění kvality služeb : Přednášky z předmětu Hardware počítačových sítí* [online]. 2009. Brno : VUT FEEC ÚTKO, [2009] [cit. 2009-12-06]. Prezentace. Text v češtině. URL: <https://www.vutbr.cz/elearning/mod/resource/view.php?id=82380>.

[16] HUCABY, David, MCQUERRY, Steve. *Konfigurace směrovačů Cisco : Autorizovaný výukový průvodce*. Jiří Veselský. 2004. autoriz. vyd. Brno : Computer Press, 2004. 632 s. ISBN 80-722-6951-8.

[17] LUDVÍČEK, Pavel. *Vliv nastavení parametrů řízení provozu na efektivnost technologie DiffServ*. Brno : VUT FEEC ÚTKO, 2009. 50 s. Vysoké učení technické v Brně. Vedoucí bakalářské práce Ing. Jiří Hošek.

[18] MOLNÁR, Karol. *Konfigurace zajištění kvality služeb (QoS) : Laboratorní cvičení z předmětu Hardware počítačových sítí* [online]. 2009. Brno : VUT FEEC ÚTKO, [2009] [cit. 2009-12-06]. Text v češtině.
URL: <https://www.vutbr.cz/elearning/mod/resource/view.php?id=83921>.

[19] MOLNÁR, Karol. *Mechanismus diferencovaných služeb* [online]. 2007. Brno : VUT FEEC ÚTKO, [2007] [cit. 2009-11-11]. Text v češtině.
URL: <http://www.utko.feec.vutbr.cz/~molnar/mmos/QoS.pdf>.

[20] PARKHURST, William. *Cisco OSPF Command and Configuration Handbook: CCIE Professional Development*. First Printing. Indianapolis: Cisco Systems, Inc, 2002. 528 s. ISBN 1-58705-071-4.

[21] POSOLDA, Marek. *QoS, klasifikace síťového provozu* [online]. 2006. MUNI FI, [2006] [cit. 2009-11-11]. Text v češtině. URL:
<http://www.fi.muni.cz/~kas/p090/referaty/2006-podzim/st/xposolda_qos.html>.

[22] SZIGETI , Tim , HATTINGH, Christina. *End-to-End QoS Network Design: Quality of Service in LANs, WANs, and VPNs*. 1st edition. [s.l.] : Cisco Press, 2004. 768 s. Networking Technology. ISBN 1-58705-176-1.

[23] UBIK, Sven. *Qos a DiffServ : Úvod do problematiky* [online]. 2000. Praha : CESNET, 2000 [cit. 2009-11-11]. Text v češtině.
URL: <http://staff.cesnet.cz/~ubik/publications/2000/diffserv.pdf>.

[24] VELTE, Toby, VELTE, Anthony. *Síťové technologie Cisco : Velký průvodce*. David Krásenský. 2003. autoriz. vyd. Brno : Computer Press, 2003. 759 s. ISBN 80-7226-857-0.

SEZNAM POUŽITÝCH ZKRATEK

ACL	Access Control List
ATM	Asynchronous Transfer Mode
BA	Behaviour Aggregate
Bc	Burst Size
CAC	Connection Admission Control
CBS	Committed Burst Size
CBWFQ	Class-Based Weighted Fair Queuing
CBWFQ	Class-Based Weighted Fair Queuing
CIR	Committed Information Rate
CLI	Command-Line Interface
CQ	Custom Queuing
CU	Currently Unused
DiffServ	Differentiated Services
D-ITG	Distribued Internet Traffic Generator
DNS	Domain Name System
DS	Differentiated Service
DSCP	DiffServ Code Point
DWRED	Distributed Weighted Random Early Detection
EBS	Excess Burst Size
ECN	Explicit Congestion Notification
FIFO	First-In First-Out
GTS	Generic Traffic Shaping
ICMP	Internet Control Message Protocol
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IntServ	Integrated Services
IP	Internet Protocol
IPv4	Internet Protocol Version 4
IPv6	Internet Protocol Version 6
ISP	Internet Service Provider
ISR	Integrated Services Router
ITU	International Telecommunication Union
LLQ	Low Latency Queuing
MAC	Media Access Control
MDRR	Modified Deficit Round Robin
MF	Multi-Field Classification
MPLS	Multiprotocol Label Switching
MQC	Modular QoS Command-Line Interface
NTP	Network Time Protocol

OSPF	Open Shortest Path First
OWD	One Way Delay
PBS	Peak Burst Size
PHB	Per-Hop Behavior
PIR	Peak Access Rate
PQ	Priority Queuing
QoS	Quality of Service
RED	Random Early Detection
RFC	Request For Comment
RSVP	Resource Reservation Protocol
SDM	Security Device Manager
SLA	Service Level Agreement
TCP	Transmission Control Protocol
TMN	Traffic-Management Node
TOS	Type Of Service
UDP	User Datagram Protocol
VoIP	Voice over IP
VPN	Virtual Private Network
WAN	Wide Area Network
WFQ	Weighted Fair Queuing
WRED	Weighted Random Early Detection