UNIVERZITA PARDUBICE
Fakulta elektrotechniky a informatiky

Principy QoS a jeho modelové nasazení

Michal Třetina

Bakalářská práce
2013

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: **2012/2013**

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Michal Třetina**
Osobní číslo:	**I10244**
Studijní program:	**B2646 Informační technologie**
Studijní obor:	**Informační technologie**
Název tématu:	**Principy QoS a jeho modelové nasazení**
Zadávající katedra:	**Katedra informačních technologií**

Zásady pro vypracování:

Cílem práce je představit principy fungování Quality of Service a možnosti jeho využití v modelové LAN síti. Na základě prostudované literatury autor podrobně představí principy fungování Quality of Service (QoS) včetně možnosti jeho využívání a nasazení. V implementační části autor navrhne a v laboratoři počítačových sítí zrealizuje minimálně tři ukázkové úlohy na využití QoS v modelové LAN síti. Pro analýzu správného nasazení QoS použije síťový analyzátor, jehož výstupy v příloze bakalářské práce budou dokazovat správnou funkci QoS na vybrané protokoly při simulovaném vytížení sítě.

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

ODOM, Wendell a Michael J CAVANAUGH. Cisco QOS exam certification guide: CCVP self-study. 2nd ed. Indianapolis: Cisco Press, c2005, xxxiv, 730 s. ISBN 15-872-0124-0.
BARREIROS, Miguel a Peter LUNDQVIST. QoS-enabled networks: tools and foundations. 1st ed. Chichester: John Wiley, 2011, xx, 222 s. Wiley series in communications networking. ISBN 978-0-470-68697-3.

Vedoucí bakalářské práce: **Mgr. Josef Horálek**
Katedra softwarových technologií

Datum zadání bakalářské práce: **21. prosince 2012**
Termín odevzdání bakalářské práce: **10. května 2013**

Univerzita Pardubice
Česká republika

L.S.

prof. Ing. Simeon Karamazov, Dr.
děkan

Ing. Lukáš Čegan, Ph.D.
vedoucí katedry

V Pardubicích dne 29. března 2013

Prohlášení autora

Prohlašuji, že jsem tuto práci vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 10. 5. 2013 Michal Třetina

Poděkování

Na tomto místě bych rád poděkoval vedoucímu bakalářské práce, Mgr. Josefu Horálkovi, za cenné rady a metodickou pomoc při jejím zpracování. Dále bych rád poděkoval své rodině za celkovou podporu ve studiu.

Anotace

Cílem práce je představit principy fungování Quality of Service a možnosti jeho využití v modelové LAN síti. Na základě prostudované literatury autor podrobně představí principy fungování Quality of Service (QoS) včetně možnosti jeho využívání a nasazení. V implementační části autor navrhne a v laboratoři počítačových sítí zrealizuje minimálně tři ukázkové úlohy na využití QoS v modelové LAN síti. Pro analýzu správného nasazení QoS použije síťový analyzátor, jehož výstupy v příloze bakalářské práce budou dokazovat správnou funkci QoS na vybrané protokoly při simulovaném vytížení sítě.

Klíčová slova

LAN, kvalita služby, QoS, integrované služby, diferencované služby, RSVP, šířka pásma, ztrátovost paketů, zpoždění, jitter, Wireshark

Title

Quality of Service and exemplary implementation

Annotation

The aim of this thesis is to introduce the principles of Quality of Service and possibilities of its deployment in an exemplary LAN. Based on the research of on-topic literature, inner workings and principles of Quality of Service (QoS) will be introduced, along with its usage and deployment. In the implementation section, the author designs and in a network laboratory implements at least 3 examples of QoS utilization in exemplary LAN. A network analyzer will be used for the assessment of correct usage of QoS and its outputs in the appendix of the thesis will confirm correct function of QoS with selected protocols during simulated load on the network.

Keywords

LAN, quality of service, QoS, integrated services, differentiated services, RSVP, bandwidth, packet loss, delay, jitter, Wireshark

Obsah

Seznam zkratek

AF	Assured Forwarding
BA	Behavior Aggregate
BSD	Berkeley Software Distribution
BW	Bandwidth
CBWFQ	Class-Based Weighted Fair Queuing
CLI	Command Line Interface
CoS	Class of Service
CU	Currently Unused
DiffServ	Differentiated Services
DS	Differentiated Services
DSCP	Differentiated Services Code Point
ECN	Explicit Congestion Notification
EF	Expedited Forwarding
EXP	Experimental
FEC	Forward Equivalence Class
FIFO	First In, First Out
FQ	Fair Queue
FTP	File Transfer Protocol
GNU	GNU's Not Unix
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
IETF	The Internet Engineering Task Force
IntServ	Integrated Services
IP	Internet Protocol
ISO	International Organization for Standardization

LAN	Local Area Network
LER	Label Edge Router
LLQ	Low Latency Queuing
LSP	Label Switched Path
LSR	Label Switched Router
MF	Multifield
MOS	Mean Opinion Score
MPLS	Multiprotocol Label Switching
OSI	Open Systems Interconnection
PHB	Per-Hop Behavior
PHP	Penultimate Hop Popping
POP	Post Office Protocol
PQ	Priority Queue
QoS	Quality of Service
RED	Random Early Detection
RFC	Request for Comments
RSVP	Resource Reservation Protocol
SIP	Session Initiation Protocol
SSH	Secure Shell
TC	Traffic Class
TCP	Transmission Control Protocol
ToS	Type of Service
TTL	Time to Live
UDP	User Diagram Protocol
VoIP	Voice over Internet Protocol
WBBRR	Weighted Bit-by-Bit Round Robin

WFQ	Weighted Fair Queuing
WRED	Weighted Random Early Detection
WRR	Weighted Round Robin

Seznam obrázků

Seznam tabulek

Úvod

Sítě IP byly navrženy pro službu „Best Effort“ – přenést data co nejkratší cestou ze zdroje k cíli a využít maximum dostupné šířky pásma. Tento návrh však nemohl předpovídat bezprecedentní růst objemu přenášených dat a další moderní požadavky na tyto sítě. Trendem ve světě sítí je dnes tzv. konvergence, kdy jedna fyzická síť přenáší rozličné typy síťového provozu. Ke slovu se dostává stále více a více aplikací, které ke své správné funkci požadují jisté záruky na kvalitu spojení – jedná se např. o multimédia, videokonference, telefonii IP a populární videohry on-line. Pokud tyto požadavky nejsou splněny, kvalita takové služby potom rychle degraduje až k praktické nepoužitelnosti. Klasická síť IP však sama o sobě postrádá mechanismy, které by mohly tyto záruky poskytnout. Problémem je také rozlišitelnost různých datových toků, neboť všechny pakety jsou si v takové síti rovny a pokud dojde k zahlcení sítě, jsou i rovnocenně zahazovány, což ovšem pro některé druhy síťových aplikací nemusí být přijatelné.

Snaha o vyřešení těchto problémů vedla k vytvoření určitých mechanismů a nástrojů QoS, které umožňují ovlivnit chování paketu dle potřeby. V důsledku tak lze zajistit lepší vyvážení zátěže sítě a spravedlivě dělit konektivitu dle nastavených priorit a vyhnout se přetížení sítě a neuspokojivé kvalitě spojení. Je důležité si ovšem uvědomit, že preferování jednoho paketu obecně negativně ovlivní paket jiný. Proto je velmi důležité tyto mechanismy důkladně prozkoumat a pochopit, neboť korektní implementaci musí předcházet poučená analýza dané situace a požadavků.

V teoretické části této bakalářské práce budou diskutovány požadavky aplikací na technologie QoS v obecné rovině a dále výše zmíněné mechanismy, které požadovanou kvalitu služby zajišťují, jako například architektura integrovaných služeb (IntServ) a architektura diferencovaných služeb (DiffServ).

V praktické části potom budou navrženy 3 testovací sítě, v nichž budou mechanismy představeny jejich konfigurací na konkrétních fyzických zařízeních. Simulací síťového provozu pomocí programu IxChariot, analýzou paketů pomocí síťového analyzátoru Wireshark a následným rozborem dat získaných při simulované zátěži bude ověřena jejich správná funkčnost a potřeba nasazení.

V závěru práce budou celkově zhodnoceny získané poznatky.

1 Obecně o kvalitě služeb

1.1 Definice

Termín kvalita služby (QoS) je dnes v oblasti sítí používán v mnoha významech. Důležité je odlišovat subjektivní vnímání kvality nějaké konkrétní síťové služby uživatelem od souboru obecných konceptů a nástrojů v sítích IP, které umožňují ovlivnit přístup paketů k síťovým prostředkům.

Podle [1] QoS definuje „schopnost sítě zajistit různé úrovně záruk na kvalitu služby pro různé druhy síťového provozu“. Síťový administrátor tak dostává do rukou prostředek, jak prioritizovat jeden druh síťového provozu před jiným a do jisté míry tak síťový provoz řídit. QoS se implementuje pro řízení zahlcené (congested) sítě, nebo naopak pro předcházení jejímu zahlcení. V prvním případě QoS pomáhá řešit situaci, kdy aplikace požadují od sítě větší šířku pásma, než ta je schopna poskytnout. Kritické a citlivé aplikace mohou dostat přednost před ostatními a uspokojivě fungovat i v prostředí zahlcené sítě. V druhém případě se její mechanismy snaží těmto situacem předcházet. Tabulka 1 uvádí možné problémy různých druhů síťového provozu bez QoS.

Tabulka 1 – Chování síťového provozu bez QoS (podle [2])

Síťový provoz	**Chování bez QoS**
Hlas	Hlasová komunikace je obtížně srozumitelná.
	Hlasová komunikace je trhaná.
	Zpoždění ztěžuje vzájemnou komunikací, účastníci hovoru neví, kdy ostatní domluvili.
	Hovory jsou násilně ukončeny.
Video	Trhaný pohyb.
	Audio není synchronizováno s videem.
	Pomalý pohyb.
Data	Data dorazí příliš pozdě, nejsou už užitečná.
	Nahodilá doba odezvy frustruje uživatele.

Důvody pro zavedení QoS lze shrnout do následujích bodů:

- Maximalizace využití sítě
- Prioritizace jednoho typu síťového provozu před jiným
- Zajištění správné funkčnosti kritických a citlivých aplikací
- Kontrola nad síťovými prostředky

Je nutno mít na paměti, že kvalita služby je závislá na všech komponentách sítě. Jednotlivé prvky sítě tedy musí QoS nějakým způsobem podporovat.

1.2 Historický vývoj

Koncept QoS není nový a prošel jistým vývojem. Již ve specifikaci IP z roku 1981 lze nalézt tzv. byte Type of Service (ToS). Dle specifikace „ToS zajišťuje indikaci abstraktních parametrů požadované kvality služby. Tyto parametry jsou použity pro výběr konkrétních parametrů skutečné služby při přenosu datagramu skrze konkrétní síť.“ [3] Internet nicméně dlouho nenacházel pro byte ToS využití, neboť objem přenášených dat byl zpočátku poměrně malý a většina implementací IP jej přehlížela. Dnes nachází uplatnění v architektuře diferencovaných služeb.

Důležitost QoS rostla s popularizací a postupnou komercializací Internetu. Internet je svou podstatou nespojovaná služba (každý paket je doručován jednotlivě a nezávisle), což mu propůjčuje značnou flexibilitu, přináší to však s sebou problém potenciálního zahlcení sítě, zejména při propojování sítí s vzájemně velmi odlišnou šířkou pásma. Tento problém diskutoval John Nagel již roku 1984 [4]. Jeho řešení, tzv. Nagleův algoritmus, bylo počátkem funkcí QoS v IP.

Další sada nástrojů QoS, vyvinutá a popsaná Van Jacobsonem roku 1988 [5], zavedla mechanismy předcházení zahlcení (Congestion Avoidance) a tzv. pomalý náběh (Slow Start) pro koncové systémy. Dnes jsou v implementacích TCP povinné. Van Jacobsonem byly dále přidány další dva mechanismy, Fast Retransmit a Fast Recovery. Fast Retransmit se spolu s Congestion Avoidance a Slow Start poprvé objevil v distribuci 4.3 BSD Tahoe roku 1988. Distribuce 4.3 BSD Reno z roku 1990 implementuje Fast Recovery. Detailnější přehled nabízí RFC 2001 [6].

Pro plnohodnotné end-to-end QoS bylo potřeba zaměřit se také na směrovače. Zde vývoj směřoval zejména k propracování technik, jak zvládat a také předcházet jejich zahlcení. Výsledkem bylo několik mechanismů, jak pakety na směrovačích řadit do front a plánovat (např. populární WFQ) a předcházet jejich zahlcování (algoritmy RED a WRED) [7]. Další vývoj směřoval ke standardizaci zajištění QoS v Internetu. Výsledkem byly architektury IntServ [8] a DiffServ [9], vše pod hlavičkou organizace The Internet Engineering Task Force (IETF).

1.3 Parametry kvality služby

Pro zajištění uspokojivé kvality služby je třeba definovat parametry, které výrazně ovlivňují výkon dané síťové aplikace. Mechanismy QoS nám pak umožní s nimi manipulovat dle konkrétních potřeb, úprava jednoho z parametrů může mít nicméně za následek zhoršení jiného. Těmito parametry jsou:

- Šířka pásma (Bandwidth)
- Zpoždění (Delay)
- Rozptyl zpoždění (Jitter)

- Ztrátovost paketů (Packet loss)

Tabulka 2 shrnuje závislost kvality aplikací na jednotlivých parametrech QoS.

Tabulka 2 – Citlivost aplikací na parametry QoS (podle [10])

Aplikace	Šířka pásma	Citlivost na:		
		Zpoždění	Jitter	Ztráty paketů
VOIP	Nízká	Vysoká	Vysoká	Střední
Video konference	Vysoká	Vysoká	Vysoká	Vysoká
Video stream	Vysoká	Střední	Střední	Střední
Audio stream	Nízká	Střední	Střední	Střední
Transakce server/klient	Střední	Střední	Nízká	Střední
Email	Nízká	Nízká	Nízká	Vysoká
Přenos souborů	Střední	Nízká	Nízká	Vysoká

1.3.1 Šířka pásma

Šířka pásma popisuje kapacitu daného média, protokolu nebo spojení při přenosu dat. Z dostupné šířky pásma se odvíjí rychlost, kterou můžeme přenášet data. Čím větší je šířka pásma, tím vyšších přenosových rychlostí lze teoreticky dosáhnout. Vyjadřujeme v bitech za sekundu (b/s). QoS ve vztahu k šířce pásma umožňuje její optimálnější využití.

1.3.2 Zpoždění

Zpoždění definujeme jako časový interval mezi odesláním dat z výchozího bodu a jeho přijetím v cíli. Udáváme obvykle v milisekudách (ms). Celkové zpoždění lze jemněji rozdělit na několik částí, neboť k nějakému zpoždění (jedno, zda pro praktické účely zanedbatelnému či nikoliv) dochází ve všech úsecích sítě. Části jsou následující [2]:

- **Propagační zpoždění (Propagation delay)** – čas, za který jeden bit projde fyzickým médiem. Závisí na rychlosti světla v daném médiu a délce média.

- **Serializační zpoždění (Serialization delay) –** čas, za který jsou jednotlivé bity paketu zakódovány a vloženy na fyzické médium. Rychlejší linky mají nižší serializační zpoždění. Kratší rámce jsou také na médium vloženy rychleji. Serializační zpoždění (t_s) lze spočítat jako:

  $$t_s = P/R$$

  kde P – je velikost paketu v bitech (b),

  R – je přenosová rychlost linky v bitech za sekundu (b/s).

- **Zpoždění ve frontách (Queuing delay) –** čas, který paket stráví ve frontách na vstupních nebo výstupních rozhraních předtím, než je přeposlán.

- **Zpoždění zpracování (Forwarding or processing delay) –** čas mezi obdržením paketu a jeho vložením do fronty paketů připravených pro přenos. Dochází k němu na všech přepínacích a směrovacích zařízeních.

Celkové zpoždění je součtem výše uvedených.

1.3.3 Rozptyl zpoždění

U paketů, které byly vyslány po řadě za sebou a do cíle dorazily s různým zpožděním, došlo k rozptylu zpoždění. Ideálně by byl tento rozdíl nulový a pakety by byly přijímány v konstatních intervalech. K určitému rozptylu nicméně dojde k síti vždy a síťové aplikace s ním počítají. Při růstu zatížení sítě často poroste i rozptyl zpoždění, což negativně ovlivňuje zejména hlasové a video služby, kde se projevuje jako poruchy spojení. Problém se složitě eliminuje pomocí nasazení vyrovnávacích pamětí, které pakety po určitou dobu pozdrží.

1.3.4 Ztrátovost paketů

Ztrátovost paketů je procentuální poměr počtu paketů, které z nějakého důvodu nedorazily do cíle, a celkového množství vyslaných paketů. Ke ztrátě paketů dochází z různých důvodů: interference v okolí cesty, kolize na linkové vrstvě, hardwarové nebo softwarové selhání (zejména ovladačů) nebo nedostatečná síla signálu v cíli komunikace. Další možnou příčinou je přetížení síťových prvků, např. přetížení procesoru nebo naplnění vyrovnávací paměti směrovače. Síťový prvek pak odhazuje pakety, dokud situace není vyřešena. Různé aplikace na ztrátu paketu reagují různě. Např. aplikace v reálném čase si nemohou ztracené pakety opětovně vyžádat, neboť by v takovém případě docházelo k neakceptovatelnému zpoždění. Obecně se však aplikace dokáží dobře vyrovnat s nízkou, nahodile rozloženou ztrátou a uživatel ji nemusí ani postřehnout. Kvalita je citelně ovlivněna zejména vysokou ztrátovostí nebo ztrátou několika po sobě následujících paketů.

1.3.5 Třídy aplikací

Aplikace lze ve vztahu k výše uvedeným parametrům dělit do poněkud obecnějších tříd [11]:

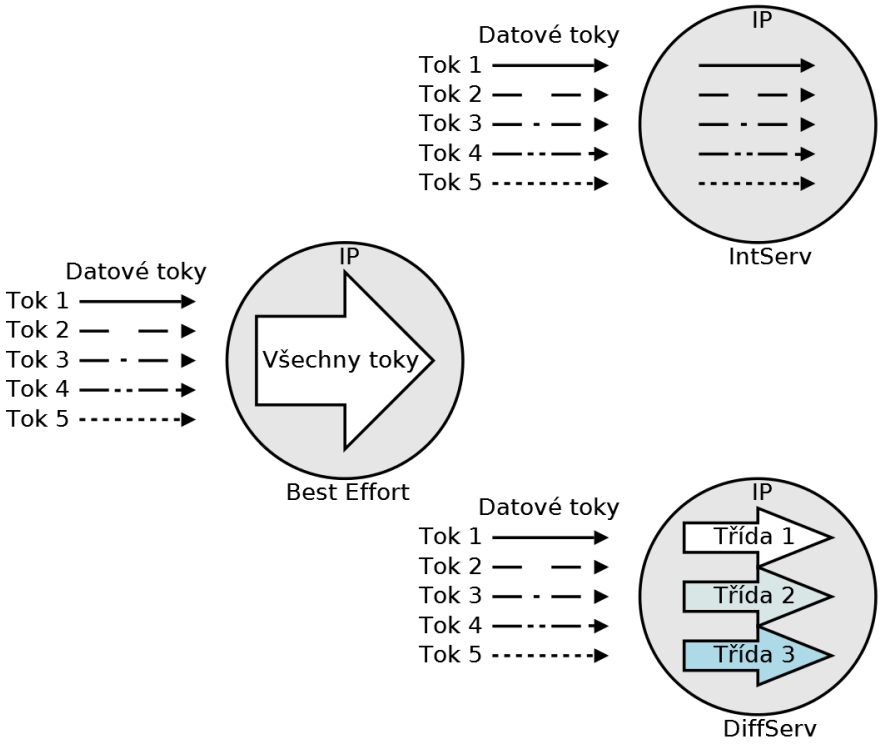
- **Elastické aplikace** – flexibilní ve svých požadavcích, mohou operovat v širokém intervalu zpoždění, ztrátovosti paketů a šířky pásma. Jde např. o Telnet, FTP, HTTP, POP, SSH a další. Na transportní vrstvě referenčního modelu ISO/OSI obvykle využívají spojovaného protokolu TCP.

- **Aplikace Real-Time Tolerant** – citlivé zejména na maximální hodnoty zpoždění. S nahodilou ztrátou paketů jsou schopny se vypořádat. Jde např. o video aplikace využívající vyrovnávácích pamětí. Z pohledu transportní vrstvy jsou postaveny nad nespojovaným protokolem UDP.

- **Aplikace Real-Time Intolerant –** nutno zajistit minimální jitter, zpoždění a ztrátovost. Jde např. o VoIP nebo videokonference. Opět využívají UDP.

2 Mechanismy kvality služeb v sítích IP

Sítě IP bez mechanismů QoS poskytují tzv. „Best Effort“ službu, kdy všechny pakety jsou si rovny a je s nimi zacházeno jedním způsobem [10]. Taková síť neposkytuje žádné záruky na parametry služeb. Mechanismy QoS umožní síti IP klasifikovat pakety či definovat datové toky a dle stanovených kritérií s nimi nakládat. Základními přístupy k zajištění QoS jsou architektury IntServ a DiffServ. Obrázek 1 ilustruje logiku jednotlivých přístupů.



Obrázek 1 – Způsoby zacházení s datovými toky pro technologie IntServ, Best Effort a DiffServ (podle [7])

2.1 Architektura integrovaných služeb

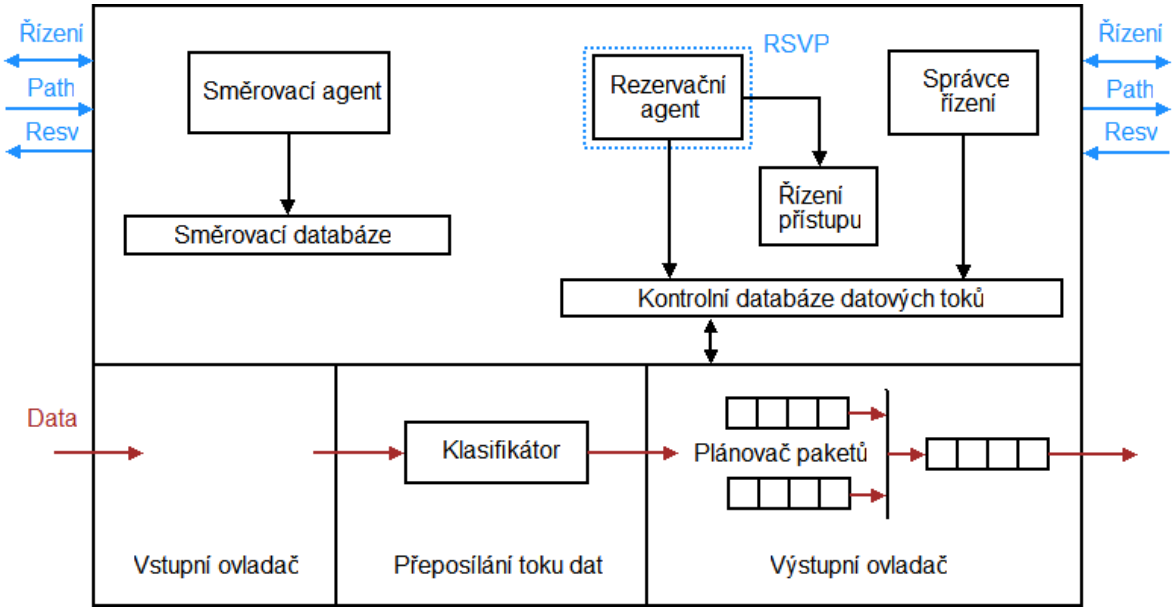
Architektura integrovaných služeb (Integrated Services, dále IntServ), definovaná v RFC 1633 [8], se stala prvním pokusem o komplexní end-to-end QoS. Hlavní myšlenkou IntServ je možnost rezervovat trasu od zdroje do cíle a vyžadovat záruky na minimální nutnou šířku pásma, zpoždění, jitter a ztrátovost paketů pro konkrétní datový tok. Datový tok v IntServ je identifikován zdrojovou a cílovou adresou IP, zdrojovým a cílovým portem a identifikátorem protokolu [7]. IntServ rozlišuje 2 volitelné modely služby:

- **Controlled Load** – vhodné pro aplikace citlivé na zahlcení sítě (Real-Time Tolerant). Výkon aplikace využívající službu Controlled Load se tak nebude zhoršovat s rostoucím zatížením sítě. Objevují se však výkyvy v hodnotách zpoždění, jitteru a ztrátovosti paketů.

- **Guaranteed Service** – designována pro aplikace Real-Time Intolerant. Aplikuje striktní hranici pro zvolený parametr, např. maximální zpoždění *x* ms.

Síť musí vykonávat řízení přístupu, tedy ověřit, zda je požadavkům možno vyhovět. Pokud ne, je takový požadavek zamítnut a kontrola vrácena zpět aplikaci, která rozhodne, jestli zkusí žádost opakovat s menšími požadavky nebo od dalších pokusů upustí. Požadavek musí být propagován všem uzlům na trase. K tomu je zapotřebí rezervačního protokolu, nejrozšířenějším je RSVP (bude diskutován dále), jeho práce je nicméně spojena se značnou režií. Jen těžko si lze představit fungování IntServ v rozsáhlých sítích (jako je Internet) s obrovským množstvím datových toků, o kterých je třeba držet stavové informace na každém směrovači [12]. Praktické využití IntServ leží spíše pouze v malých podnikových sítích.

Obrázek 2 zobrazuje referenční model architektury IntServ:



Obrázek 2 – Referenční model architektury IntServ (podle [8])

Implementační rámec IntServ je tvořen 4 hlavními komponentami:

- **Plánovač paketů** – implementován na místech, kde jsou pakety řazeny do front (např. směrovače, přepínače). Stará se o odesílání proudů paketů z výstupních portů, k čemuž využívá fronty, do kterých jsou pakety vkládány na základě klasifikace, a další mechanismy, např. časovače. Přímo tak ovlivňuje dobu, kterou paket ve frontě stráví, nebo zda do ní vůbec vstoupí (skončí odhozen).

- **Řízení přístupu** – implementuje algoritmus, který rozhoduje o přijetí nebo odmítnutí požadavku na rezervaci QoS. K odmítnutí dojde v případě, že by požadavek negativně ovlivnil předchozí rezervace. Proběhne na každém uzlu sítě.

- **Klasifikátor** – rozděluje příchozí pakety do tříd, se kterymi následně pracuje plánovač paketů.

- **Protokol pro rezervaci zdrojů** – rezervuje zdroje a udržuje stavy v síťových zařízeních na dané rezervované trase, obvykle se jedná o RSVP (nebo novější YESSIR).

2.1.1 Resource Reservation Protocol

Jelikož IntServ pracuje na bázi rezervací jednotlivých síťových prostředků a jejich správy, je třeba tyto rezervace po síti nějak šířit. Typickým signalizačním prostředkem je Resource Reservation Protocol (RSVP), definovaný v RFC 2205 [13]. RSVP lze popsat těmito klíčovými vlastnostmi:

- Příjemce datového toku iniciuje a udržuje rezervaci prostředků pro daný tok, RSVP je *receiver-oriented*.

- Je nezávislý na konkrétním použitém směrovacím protokolu.

- Je ve své podstatě simplexový, z čehož vyplývá, že pokud je nutno rezervovat v obou směrech, musí žádost o rezervaci prostředků podat oba dva koncové body komunikace.

- Rezervace jsou časované, v každém směrovači je nastaven interval, po jehož vypršení je nutno provést periodický update. Pokud update není proveden, rezervace je zrušena. Takový přístup nazýváme *soft*, měkký.

- Možnost zvolit model služby jako Guaranteed Service nebo Controlled Load.

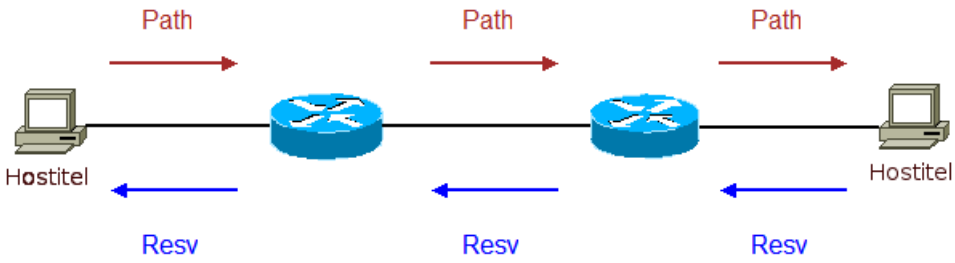
- Rozlišuje 3 typy rezervací, které definují způsob, jakým jsou spojovány požadavky: Shared Explicit (tvoří jednu sdílenou rezervaci pro několik vybraných odesilatelů), Wildcard-Filter (jedna sdílená rezervace pro více příjemců, kde velikost rezervace se rovná největšímu z požadavků) a Fixed-Filter („jednotlivé rezervace a explicitní volba odesílatelů“ [13], každý si tvoří vlastní rezervaci)

Dále si RSVP definuje 7 typů zpráv, které pro své potřeby využívá, viz tabulka 3. Rozlišují se dle osmibitového pole v hlavičce zprávy protokolu.

Tabulka 3 – Typy RSVP zpráv (podle [13])

Číslo zprávy	Typ zprávy	Význam zprávy
1	Path	Navázání spojení
2	Resv	Potvrzení spojení, rezervace prostředků
3	PathErr	Chyba poslední Path zprávy
4	ResvErr	Chyba poslední Resv zprávy
5	PathTear	Vymaže korespondující Path stav
6	ResvTear	Vymaže korespondující Resv stav (rezervaci)
7	ResvConf	Potvrzuje rezervační požadavky, odpověď na Resv

Pro samotnou funkci RSVP jsou zásadní zejména zprávy typu Path a Resv. Rezervace typicky začíná zasláním zprávy Path odesílatelem příjemci. Každý uzel na trase si vytvoří stav Path. Pokud není schopen dostát požadavkům, šíří zpět zprávu PathErr. Když Path dorazí do cíle, příjemce odpoví zprávou Resv, která nese požadované QoS charakteristiky, vrátí se po stopách zprávy Path a rezervuje prostředky. Pokud by rezervace nebyla možná, šíří se zpráva ResvErr. Úspěšně navázané spojení potvrdí zpráva ResvConf a začne přenos dat. Zprávy PathTear a ResvTear slouží k jeho ukončení. Pokud mezi směrovači využívajícími RSVP leží prostředí bez implementovaného RSVP, je protokol tunelován. V takovém případě však nemůže být poskytována plná end-to-end záruka. Navázání spojení zachycuje obrázek 3.



Obrázek 3 – Navázení spojení pomocí RSVP

2.2 Architektura diferencovaných služeb

Jako odpověď na potenciální problém (potenciální proto, že architektura IntServ se nedočkala v Internetu praktického nasazení) škálovatelnosti architektury IntServ sestavila organizace IETF pracovní skupinu, která v RFC 2475 roku 1998 definovala architekturu diferencovaných služeb (Differentiated Services, dále DiffServ) [9].

Jedná se o implementačně jednodušší a na prostředky méně náročné řešení, které stojí v pomyslném středu mezi modely Best Effort a IntServ. Podle pole DS v hlavičce paketů IP jsou pakety agregovány do definovaných tříd. Zásadním rozdílem oproti IntServ je to, že nejsou tvořeny žádné rezervace a udržovány stavy, odpadá tedy režie s nimi spojená. Pouze jsou pakety na hranici tzv. domény DiffServ označeny a v jejím rámci síťovými prvky vzhledem k označení definovaným způsobem zpracovány. Domény DiffServ jsou dalším rozdílem oproti modelu IntServ, který vždy pracuje jako end-to-end. Pro praktické nasazení DiffServ je dobré si uvědomit, že neposkytuje pevné záruky dynamicky řízené konkrétní aplikací ve stylu IntServ, nýbrž záruky třídám provozu a jejich vzájemné vztahy (např. přednost jedné třídy před jinou) jsou předem staticky konfigurované. Pro určité situace může být vhodnější volbou IntServ.

2.2.1 Pole Differentiated Services

Pole Differentiated Services (DS) je základem pro klasifikaci paketu. Uloženo je na místě 8 bitů dlouhého pole v hlavičce paketu IPv4 zvaného Type of Service (ToS) a efektivně jej tak nahrazuje pro potřeby modelu DiffServ. Struktura viz obrázek 4. Prvních 6 bitů slouží

pro identifikátor Differentiated Services Code Point (DSCP), který tedy může nabývat 2^6 hodnot. Mapování hodnoty DSCP na třídy závisí na implementaci. Zbylé 2 bity pole se zatím nepoužívají – Currently Unused (CU).

Obrázek 4 – Pole DS

V IPv6 je tato informace zanesena v poli Traffic Class (TC).

2.2.2 Per-Hop Behavior

Per-Hop Behavior (PHB) lze definovat jako chování síťového prvku vůči paketům v doméně DiffServ. Specifické chování pak souvisí s konkrétní hodnotou DSCP a je implementováno pomocí front, jejich správy a plánovaného odesílání paketů. K plánovanému odesílání paketů dochází na každém odchozím rozhraní směrovačů. Nutno podotknout, že plánované odesílání paketů není standardizované a závisí tak na typu zařízení a výrobci [7]. Fronty jsou rozbrány v samostatné podkapitole 2.2.5.

IETF definuje 4 standardní třídy PHB + třídu default.

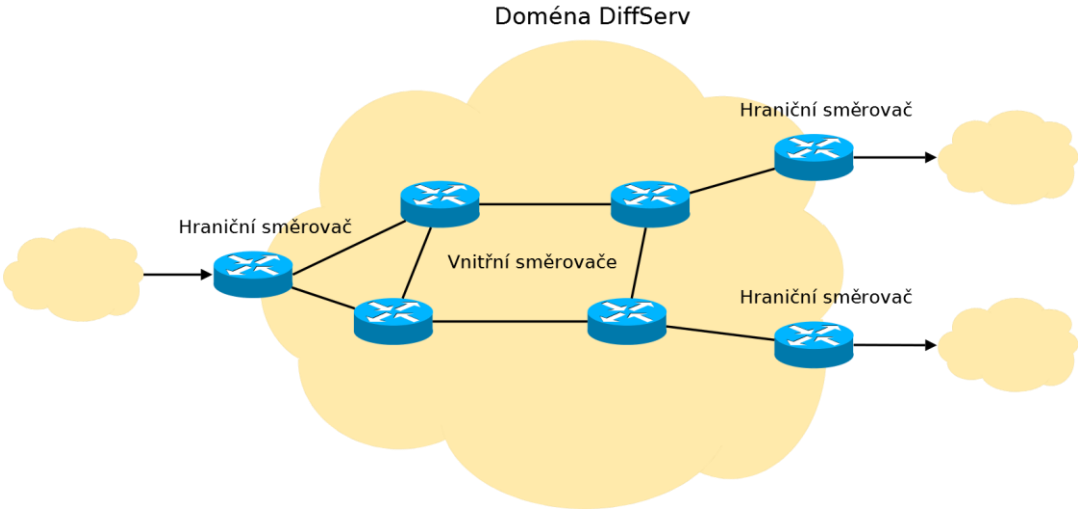
- **Default PHB** – pro síťový provoz, který nepatří do ostatních tříd. Doporučený formát DSCP je 000000_B.
- **Class Selector PHB** – pro zpětnou kompatibilitu se sítěmi, pracujícími s již zastaralou definicí ToS, kde první 3 bity tvořily tzv. IP precedenci. Formát DSCP je $xxx000_B$, kde *x* nabývá hodnoty 1 nebo 0. Třída je definována v RFC 2474 [14].
- **Expedited Forwarding PHB** – pro služby nárokující nízkou ztrátovost paketů, jitter a zpoždění, typicky aplikace v reálném čase (VoIP). Prakticky se chová jako virtuální pevný okruh. Doporučený formát DSCP je 101110_B. Třída je definována v RFC 3246 [15].
- **Voice Admit PHB –** stejné vlastnosti jako Expedited Forwarding PHB, třída je však také přijímána sítěmi, které využívají Call Admission Control. Doporučený formát DSCP je 101100_B (0). Definována v RFC 5865 [16].
- **Assured Forwarding PHB –** cílem je maximálně spolehlivé doručení paketu, jitter a zpoždění zde nehrají roli. Pakety si dále dělí do 4 vnitřních tříd, kdy vyšší třída má garantovánu větší šířku pásma. V každé ze tříd navíc rozeznáváme 3 pravděpodobnosti (nízkou, střední a vysokou) pro odhození paketu v případě, kdy datový tok překročí vyhrazené zdroje své třídy. Doporučený formát DSCP v rozsahu 001010_B – 100110_B. Definována v RFC 2597 [17].

2.2.3 Doména diferencovaných služeb

Takzvaná doména DiffServ je samosprávná jednotka architektury diferencovaných služeb. Prakticky jde o skupinu směrovačů sdílejících jednotnou politiku DiffServ. Obrázek 5 uvádí příklad takové domény. Několik souvislých domén DiffServ tvoří tzv. oblast DiffServ. Směrovače dělíme v rámci domény do dvou skupin [9]:

- **Hraniční směrovač** – leží na hranici domény, kde sousedí s další doménou DiffServ nebo jiným úsekem sítě. Provádí klasifikaci a značkování paketů. Lze dále dělit na vstupní (Ingress) a výstupní (Egress) směrovače, dle toho, jakým směrem pakety dále putují. Vstupní směrovač označí pakety vstupující do domény, výstupní značku odebírá, pakety opouští doménu. Směrovač spojující dvě domény pracuje jako vstupní pro jednu a výstupní pro druhou z domén.

- **Vnitřní směrovač –** směrovač, který není hraničním. Leží uvnitř domény a pouze přeposílá označené pakety.

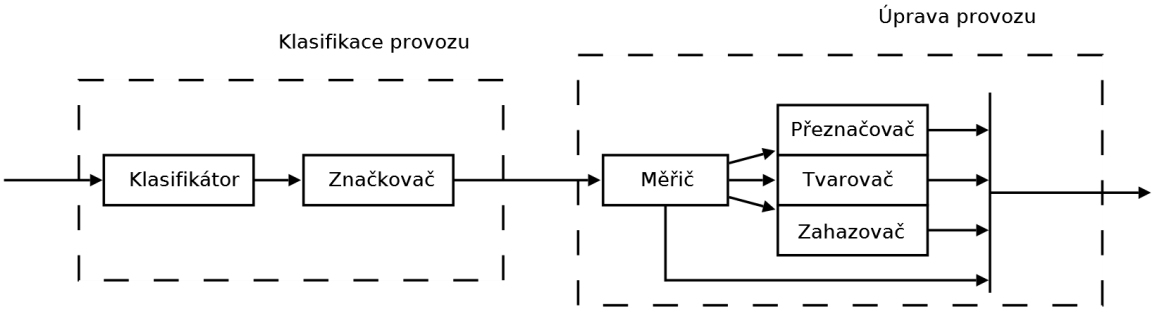


Obrázek 5 – Ukázková topologie domény DiffServ

S doménami DiffServ také souvisí pojem Service Level Agreement (SLA), tedy ujednaná úroveň služeb, kterou může zákazník v doméně očekávat. SLA může být aplikovaná i na rozhraní domén DiffServ různých poskytovatelů připojení. SLA jasně definuje konkrétní podmínky síťového provozu.

2.2.4 Referenční model architektury diferencovaných služeb

Schéma referenčního modelu představuje obrázek 6. Popsán v RFC 2475 [9], referenční model DiffServ definuje jisté základní bloky, jejichž činnost probíhá na hraničních směrovačích. Tyto bloky lze co do společné funkce dělit na dva hlavní, tzv. blok klasifikace provozu a blok úpravy provozu. Ty lze podrobněji dělit do následujících modulů [18]:



Obrázek 6 – Referenční model architektury DiffServ (podle [18])

- **Klasifikátor (Classifier) –** dělí příchozí pakety do skupin dle předdefinovaných pravidel. Rozlišujeme dva typy klasifikátorů: behavior aggregate (BA), který dělí výhradně dle hodnoty DSCP v hlavičce, a multifield (MF), který jako identifikátor používá libovolnou kombinaci zdrojové adresy, cílové adresy, zdrojového portu, cílového portu a ID protokolu.

- **Značkovač (Marker)** – po roztřídění jsou paketům značkovačem přiřazeny značky DSCP, identifikující tak jejich třídu a definující PHB.

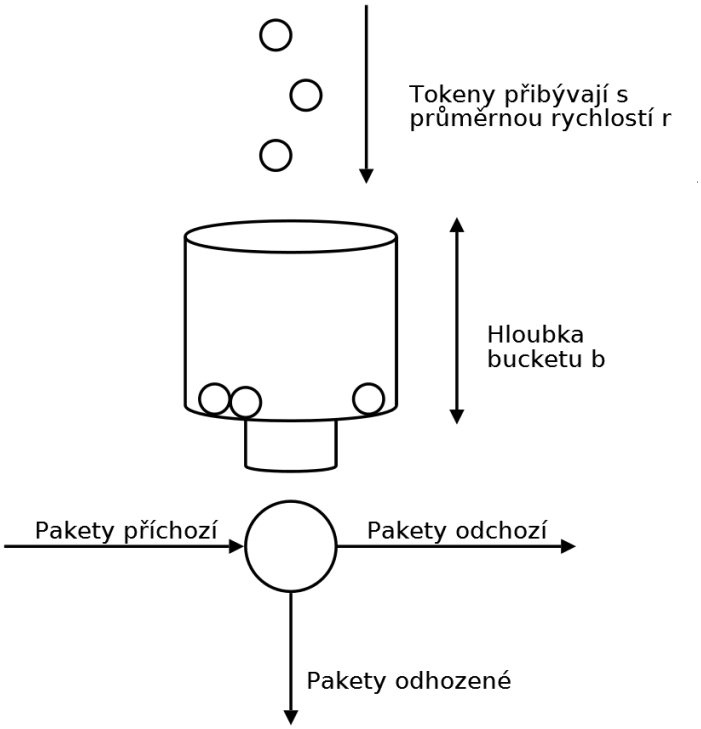
- **Měřič (Meter)** – hlídá síťový provoz každé třídy a porovnává jeho parametry s datovými profily. Síťový provoz v rámci profilu je vpuštěn dále do sítě, provoz nad rámec profilu je dále zpracováván přeznačovačem, tvarovačem nebo zahazovačem. Měřič se obvykle implementuje jako token bucket, schéma viz obrázek 7.

  Základem algoritmu token bucket je abstraktní nádoba o hloubce *b*, která obsahuje jistý počet tokenů, které do nádoby přibývají s průměrnou rychlostí *r*. Příchozí paket je pak vpuštěn dále do sítě pouze v případě, že je k dispozici množství tokenů minimálně odpovídající jeho velikosti. Pokud je tato podmínka splněna, je paket odeslán a zároveň z nádoby odebrán odpovídající počet tokenů. V opačném případě je paket pozdržen nebo dokonce odhozen.

- **Zahazovač (Dropper)** – pakety, které nedodržují rámec datového profilu, zahazuje. Totéž nastane, pokud se naplní vyrovnávací paměť tvarovače.

- **Přeznačovač (Re-marker)** – přeznačuje již jednou označené pakety. Provoz je přeznačován na rozhraní dvou domén DiffServ nebo tehdy, pokud překračuje rámec profilu.

- **Tvarovač (Shaper)** – ve vyrovnávací paměti pozdržuje pakety nedodržující rámec datového profilu. Jedná se tedy o silnější formu správy provozu než u přeznačovače.

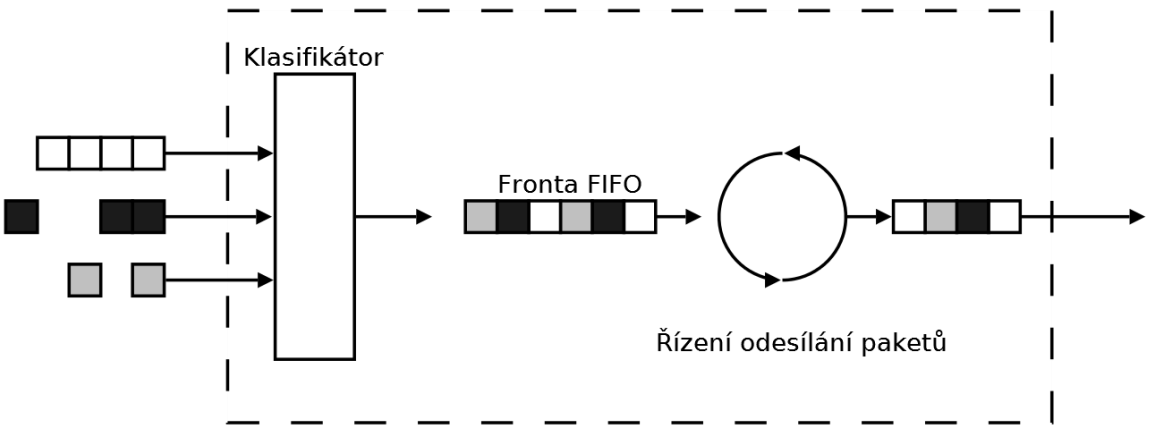


Obrázek 7 – Token Bucket (podle [1])

2.2.5 Fronty

Roztříděné pakety jsou v paměti na síťových prvcích rovnány do jednotlivých front, kde je s nimi dle typu fronty dále nakládáno. Fronty dělíme následovně [7]:

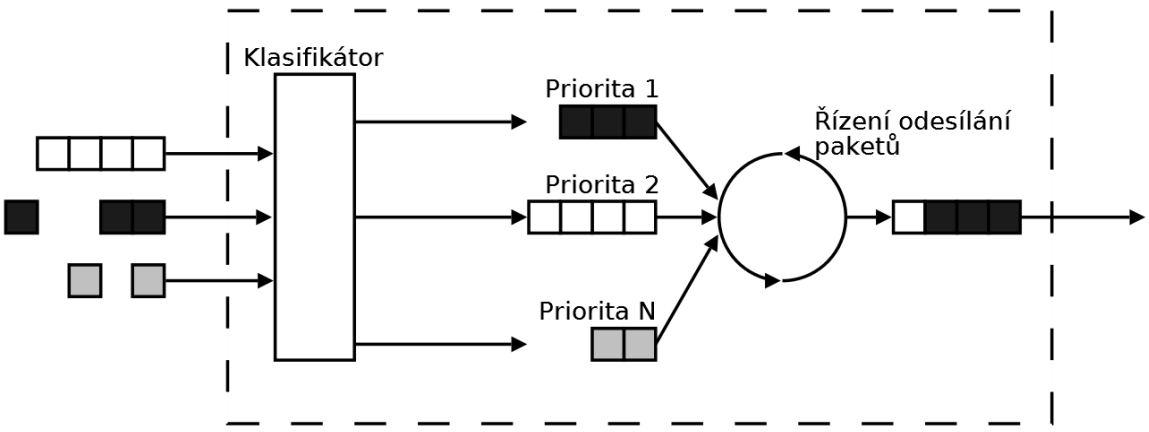
- **First In, First Out (FIFO) –** nejjednodušší metoda, kdy pakety řadíme dle času jejich příchodu. První příchozí paket opouští frontu jako první. Jedná se o implicitní způsob řazení paketů v síťových prvcích, který se používá pokud nemáme nastaven některý z pokročilejších způsobů. Nelze nijak prioritizovat pakety, neumožňuje žádné nastavení QoS a přenos je tak vlastně Best Effort. Výhodou je snadná implementace. Obrázek 8 předvádí princip FIFO.



Obrázek 8 – FIFO

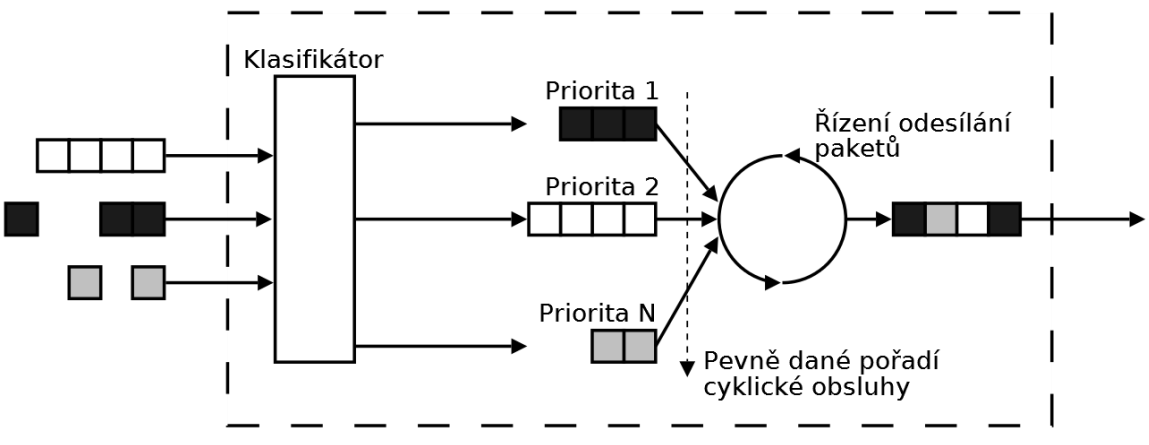
- **Prioritní fronta (PQ)** – umožňuje již práci s prioritou provozu. Základem je několik front typu FIFO, kdy každá má přiřazenou nějakou prioritu. Přednostně jsou pak odebírány pakety z fronty s vyšší prioritou, dokud ta není prázdná.

  Výhodou je opět snadná implementace, nevýhodou potenciální uváznutí, kdy pakety z front s nižší prioritou nemusí být nikdy obslouženy a budou se jevit jako ztracené. Je tedy vhodné, aby fronty s vysokou prioritou byly využívány pouze minoritním a kritickým síťovým provozem. Obrázek 9 zachycuje princip PQ.



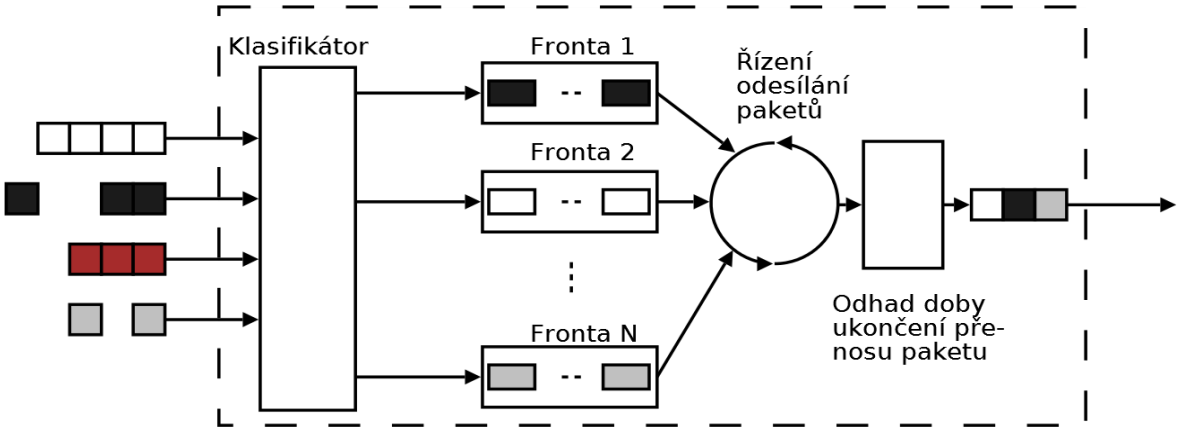
Obrázek 9 – Prioritní fronta

- **Fronty se spravedlivou obsluhou (FQ)** – řadí pakety do *N* front a každé z front připadá 1/*N* kapacity výstupního portu, kam jsou pakety posílány cyklicky z každé z neprázdných front. Někdy se tento způsob také nazývá round-robin. Tato fronta opět nabízí snadnou implementaci. Nevýhodou je to, že nezohledňuje velikosti jednotlivých paketů, a pevné dělení šířky pásma jednotlivými frontami, přestože různé datové toky mají různé nároky. Obrázek 10 představuje schéma FQ.



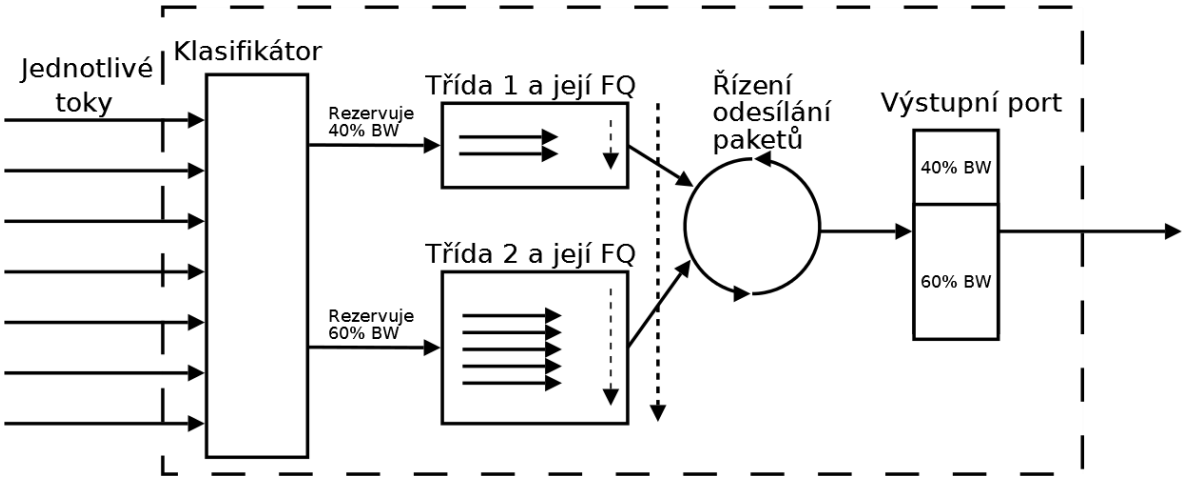
Obrázek 10 – FQ

- **Fronta s váženou spravedlivou obsluhou (WFQ)** – adresuje problém FQ s různou velikostí jednotlivých paketů. Síťový provoz je dělen do *N* front, kde každé z front je přidělena váha w_i, která vyhrazuje určitou šířku pásma. Součet všech vah je vždy 1. Narozdíl od FQ, kde jsou vždy posílány celé pakety, plánovač WFQ cyklicky navštěvuje jednotlivé fronty a odebírá jednotlivé bity. V momentě, kdy je z nich složen celý paket, je vyslán. Delší pakety tak logicky čekají na své složení déle. Tento teoretický model se nazývá WBBRR (Weighted Bit-by-Bit Round Robin). Prakticky však WFQ pouze počítá okamžik, kdy by bylo bit-by-bit posílání paketu ukončeno, a v souladu s výsledkem tohoto výpočtu jej naplánuje. Tento systém je značně výpočetně i implementačně náročný. Schéma viz obrázek 11.



Obrázek 11 – WFQ

- **Fronta s váženou cyklickou obsluhou (WRR)** – řeší druhý problém FQ, tedy pevné dělení šířky pásma. Tento způsob umí vyhradit podíl šířky pásma dle váhy w_i dané třídy, která jistým způsobem reprezentuje nároky na šířku pásma. Součet vah vždy dává 1 (100 %). Počet tříd je *N* a jednotlivé třídy obsahují vícečetné fronty. Nedochází zde již k fixnímu dělení jako u FQ, mechanismus FQ je nicméně stále využíván pro obsluhu uvnitř tříd.



Obrázek 12 – WRR

Plánování paketů probíhá jako round-robin ve dvou stupních: nejdříve jsou cyklicky obsluhovány jednotlivé třídy a v rámci tříd opět cyklicky jednotlivé fronty.

- **Fronta s váženou cyklickou obsluhou s řízením podle tříd (CBWFQ) –** mechanismus je svou podstatou podobný WRR, uvnitř tříd (ty jsou uživatelsky definované, např. na základě hodnoty DSCP) se však pro obsluhu místo FQ používá WFQ, je však možné i použití FIFO. CBWFQ dokáže poskytnout záruky na šířku pásma, nikoliv však už snížit zpoždění či jitter.

- **Low Latency Queuing (LLQ) –** jedná se o rozšíření CBWFQ přidáním podpory prioritní fronty. Lze do ní umístit vybraný síťový provoz (typicky hlas a video) a odesílat jej přednostně. Lze tak oproti CBWFQ poskytnout záruky na zpoždění a jitter a provádět omezování (policing) provozu. Více o omezování provozu v podkapitole 2.3.

2.2.6 Správa front

Fronty popsané v předchozí podkapitole 2.2.5 mají pevně danou délku. Situaci, kdy se taková fronta zaplní, je možno řešit dvěma způsoby. Prvním je tzv. pasivní správa fronty, kdy se využívá mechanismu tail-drop a bez rozdílu se odhazují nově příchozí pakety, dokud stav zaplnění fronty trvá. Tail-drop je poněkud problematickým pro služby využívající TCP, neboť kombinace schopnosti TCP reagovat na zahlcení sítě a opakovaného vysílání ztracených segmentů může vést k tzv. globální synchronizaci TCP, která snižuje efektivitu síťových zdrojů díky velkému množství paketů, které musí být odhozeny a znovu odeslány [19]. Řešením je tzv. aktivní správa front, která se snaží potenciálnímu zahlcení předcházet. Rozlišujeme dva základní algoritmy:

- **Random Early Detection (RED) –** předchází zahlcení náhodným zahazováním paketů před zaplněním front. Směrovač monitoruje množství paketů ve frontě a v momentě, kdy množství překročí administrativně nastavenou hranici, začne provádět náhodná odhození. Intenzita odhazování se pak zvyšuje se stoupajícím zaplněním fronty. TCP na toto odhazování reaguje snížením rychlosti odesílání segmentů. Nevýhodou je, že se RED chová ke všem datovým tokům stejně a navíc ji nelze uplatnit pro aplikace využívající UDP, neboť ty nejsou schopny řídit rychlost odesílání datagramů.

- **Weighted Random Early Detection (WRED) –** rozšíření mechanismu RED o možnost diferencovat jednotlivé datové toky dle jejich třídy a nastavit různé profily pro jejich odhazování. Pakety s různou klasifikační značkou pak mají různou šanci na to být odhozeny.

2.3 Omezování a tvarování provozu

Omezování (policing) a tvarování (shaping) síťového provozu jsou další nástroje, které umožňují síťovému administrátorovi ovlivňovat kvalitu služby. Poněkud se liší od technik

pro management a předcházení zahlcení tím, že negarantují minimální šířku pásma, nýbrž pouze jeho využitelné maximum pro konkrétní službu. Pro svou činnost využívají mimo jiné algoritmu Token Bucket nebo jeho varianty Dual Token Bucket se dvěma abstraktními nádobami. Rozdíl mezi omezováním a tvarováním je v tom, jak se chovají k provozu, který překročí dohodnuté pásmo. Omezování odhazuje veškerou komunikaci, která povolené pásmo překročí, kdežto techniky tvarování se snaží nárazový tok rozložit do delšího intervalu, obvykle pozdržením paketů s nižší prioritou. K algoritmu Token Bucket tak tvarování potřebuje ještě frontu pro čekající pakety. Tvarování je ve výsledku méně striktní než omezování.

2.4 Porovnání architektur integrovaných služeb, diferencovaných služeb a Best Effort

Tabulka 4 nabízí přehledné srovnání jednotlivých modelů.

Tabulka 4 – Srovnání modelů QoS

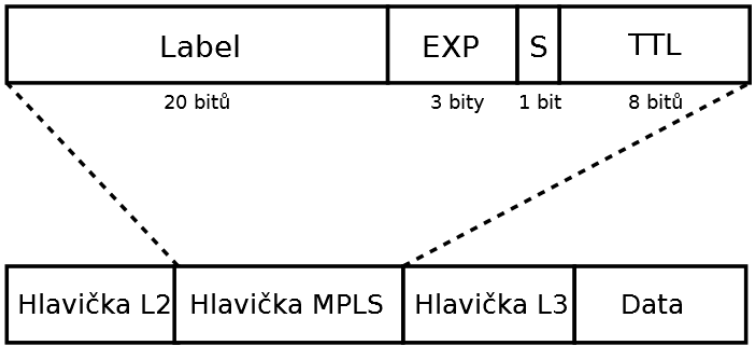
Architektura	Výhody	Nevýhody
Best Effort	Jednoduchost. Netřeba konfigurace. Bez režie.	Bez záruky na kvalitu služby. Nerozlišuje datové toky.
IntServ	Jemná kontrola jednotlivých datových toků. Poskytuje pevné záruky.	Problém škálovatelnosti ve velkých sítích. Režie rezervačního protokolu. Odesílatel, příjemce i směrovače na trase musí podporovat identifikaci dat, toků a plánování paketů.
DiffServ	Méně náročný než IntServ. Flexibilní implementace. Separace různých druhů síťového provozu.	Poskytované záruky nelze dynamicky řídit jako u IntServ.

2.5 Přepojování paketů s návěstím

Technologie Multiprotocol Label Switching (MPLS) se, jak už název napovídá, zabývá přepojováním paketů [20]. Datové toky mezi dvěma body jsou v síti IP standardně děleny do jednotlivých paketů, každý s vlastní hlavičkou obsahující adresu IP zdroje a cíle. Místo náročné práce s adresami IP a jejich vyhledáváním ve směrovacích tabulkách však MPLS pracuje s vlastními 32bitovými návěstími, které jsou paketu přiděleny na hranici domény MPLS (diskutovány dále). Obrázek 13 ilustruje podobu návěstí MPLS.

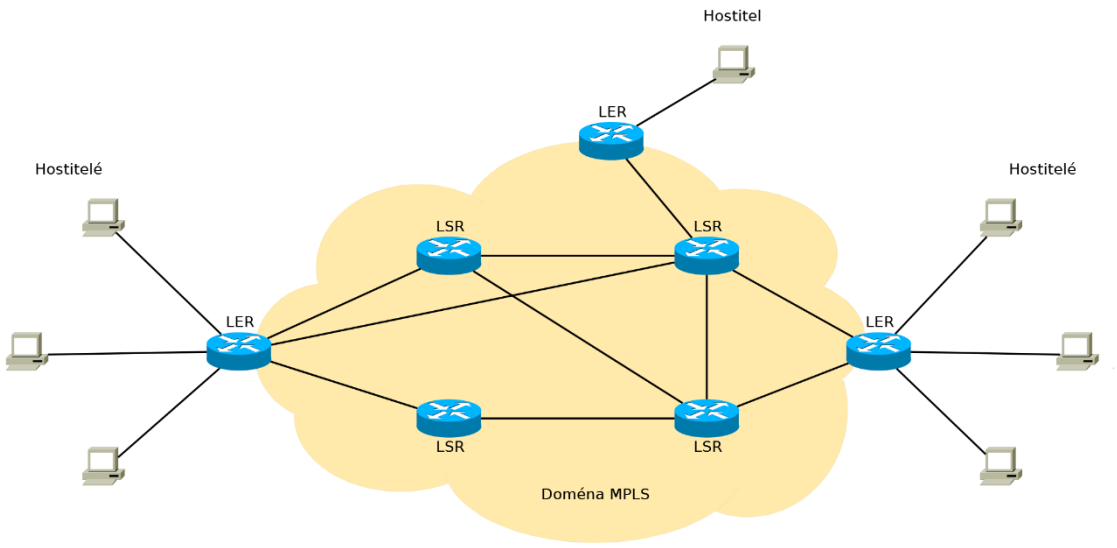
- **Label** – hodnota návěstí, na jejímž základě probíhá samotné přepojování paketů.
- **EXP (Experimental)** – původně experimentální pole, někdy využívané pro QoS, v RFC 5462 [21] bylo navrženo přejmenování pole na Traffic Class.

- **S (Bottom of Stack)** – identifikátor hlavičky rámce MPLS zapouzdřeného v rámci MPLS jiném.
- **TTL (Time to Live)** – pro předcházení smyčkám.



Obrázek 13 – Návěstí MPLS (podle [7])

Hlavní myšlenkou MPLS je pomocí návěstí sdružovat datové toky, neboť zkoumání všech hlaviček paketů IP v jednotlivých logických datových tocích je plýtvání prostředky. Směrovače uvnitř domény MPLS pak již u označených paketů nemusí odstraňovat hlavičku linkové vrstvy a směrují je pouze dle návěstí bez komplexního vyhledávání ve směrovacích tabulkách, čímž se směrování stává rychlejším. Pro distribuci návěstí mezi směrovači slouží Label Distribution Protocol (LDP) [22], nebo jiný. Veškerá činnost MPLS probíhá výhradně na směrovačích, pro koncové stanice je MPLS plně transparentní. Nad sdruženými datovými toky lze navíc aplikovat QoS využíváním dedikovaných tras pro jednotlivé agregované třídy datových toků. MPLS pracuje mezi linkovou a síťovou vrstvou ISO/OSI modelu a nezávisí na použité technologii linkové vrstvy (jako je např. Ethernet, ATM, Frame Relay). V literatuře bývá protokol MPLS často označován jako protokol vrstvy 2,5.



Obrázek 14 – Doména MPLS

Činnost MPLS probíhá v tzv. doméně MPLS, kterou definujeme jako uzavřenou síť směrovačů, které rozumí protokolu MPLS. Obrázek 14 představil příklad topologie jedné takové domény. Na vstupu do domény se nachází směrovače Label Edge Router (LER), které příchozím paketům přiřazují návěstí a na výstupu jej odstraňují. Samotné přepojování v doméně provádí směrovače Label Switched Router (LSR). Trasu, po které se paket v doméně vydá, nazýváme Label Switched Path (LSP). Tyto jednosměrné virtuální trasy jsou předem definovány – LER zařadí na vstupu paket do jedné z tříd Forward Equivalence Class (FEC) a podle třídy je dále přidělena konkrétní hodnota návěstí, dle které jsou na LSR pakety směrovány. Volba FEC může sloužit v důsledku také jako nástroj QoS, kdy určité datové toky/třídy provozu mohou mít své exkluzivní trasy. Tyto metody dále rozpracovává technologie MPLS Traffic Engineering. Je nutno poznamenat, že MPLS nedefinuje novou architekturu QoS. Implementace MPLS QoS v sítích IP často využívají modelu DiffServ s garancí síťových prostředků pro jednotlivé třídy provozu.

3 Postup praktické implementace kvality služeb

3.1 Softwarové vybavení

Pro analýzu jednotlivých paketů, zejména značení síťového provozu v rámci jednotlivých sítí navržených a realizovaných v následujících praktických úlohách byl nasazen program Wireshark [23]. Jedná se o multiplatformní open source software dostupný pod GNU General Public License v2. Nabízí podporu offline i online hloubkové analýzy velkého množství protokolů, různých typů sítí a kvalitní grafické uživatelské rozhraní. Dále byl použit profesionální nástroj IxChariot, který umožňuje simulovat rozličné scénaře síťového provozu v dané síti a online analýzu parametrů sítě [24]. Průběh analýzy a její výsledky lze přehledně zobrazit na obrazovku.

3.2 Hardwarové vybavení

Pro výstavbu modelových topologií byly použity směrovače Cisco 2811 Integrated Services Router (na obrázku 15) s operačním systémem Cisco IOS ve verzi 12.4. Tato kombinace umožnuje pohodlnou a kvalitní konfiguraci QoS pomocí rozhraní příkazové řádky (CLI).

Obrázek 15 – Cisco 2811 Integrated Services Router

Podrobnější specifikace zařízení Cisco 2811 Integrated Services Router viz [25].

3.3 Metodika měření

Před samotným měřením byla vždy nejdříve fyzicky zapojena daná topologie sítě pomocí kabelů UTP kategorie 5e pro spojení hostitel-směrovač, Smart Serial V.35 pro spojení mezi jednotlivými směrovači a na hostitelích byl nainstalován potřebný software. Hostitelé spolu komunikovali pomocí softwarových agentů IxChariot Endpoint, které jsou součástí programu IxChariot, ale lze je instalovat i samostatně. Umožňují vytvořit tzv. *páry endpoint* a slouží jako koncové body pro datové toky, které budou simulovat zatížení sítě. Typicky se na jednoho z hostitelů nainstaluje kompletní program IxChariot s ovládáním

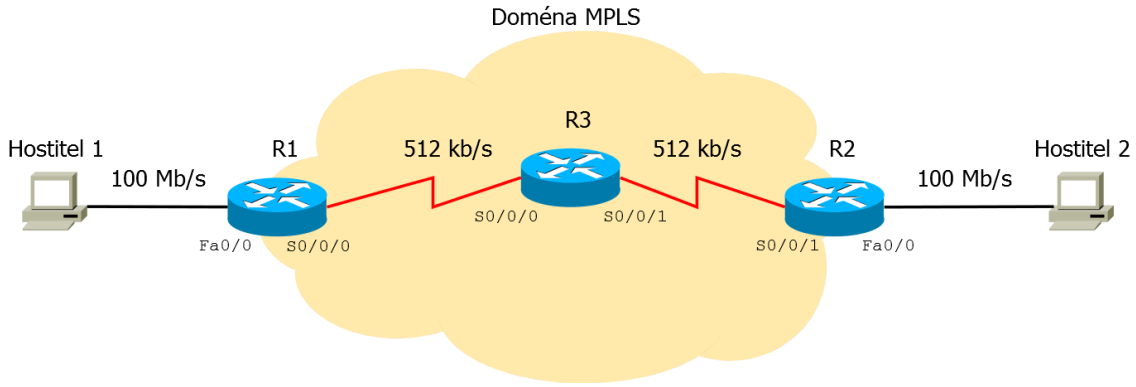
generování síťového provozu a grafickým uživatelským rozhraním a na hostitele, ke kterým bude síťový provoz směřovat, již jen samotný IxChariot Endpoint. Mezi nimi pak bude probíhat komunikace. Jelikož v testovacích scénařích bylo prováděno také značení a přeznačovaní pole DS jednotlivých paketů, bylo jeho korektnost třeba prokázat pomocí programu WireShark.

V takto připravených sítích byla poté provedena měření parametrů sítě (zpoždění, jitter atd.) nejdříve bez spuštěných nástrojů QoS (se všemi toky bylo zacházeno výhradně metodou Best Effort) a následně již s nástroji QoS plně nakonfigurovanými a spuštěnými. Doba trvání měření byla vždy nastavena na 5 minut. Veškerý síťový provoz a podmínky byly čistě laboratorní, kdy zjevnou výhodou je naprostá kontrola nad sítí. Středem pozornosti byla určena zejména hlasová komunikace VoIP, provoz FTP a videa je chápán spíše jako způsob, jak vhodně simulovat prostředí zatížené (resp. zahlcené) síťě a při konfiguraci QoS jsou právě pro hlasovou komunikaci VoIP vytvořeny nejlepší podmínky. Cílem všech měření bylo vždy potvrdit, zda implementace QoS v dané síti opravdu zlepší parametry kvality služby, tj. ztrátovost paketů, zpoždění, jitter a také tzv. odhad MOS (diskutován dále). V závěru každého scénáře pak je provedeno porovnání před a po implementaci QoS. Podrobnější informace o ad hoc nastavení jednotlivých scénářů jsou vždy uvedeny u konkrétního scénáře.

4 Testovací scénáře kvality služeb

4.1 Kvalita služby a přepojování paketů s návěstím

V prvním scénáři praktické části práce bude implementace QoS provedena v síti s doménou MPLS. Obrázek 16 představuje podobu testovací topologie. Síťový provoz bude generován na hostiteli 1 a poté směrován od hostitele 1 k hostiteli 2. Podrobné nastavení adresace IP sítě je potom k dispozici v příloze A.1 bakalářské práce.



Obrázek 16 – Topologie scénaře č. 1

Následuje ukázka nastavení směrovače pomocí CLI na příkladu směrovače R1.

```
Router> enable
Router# configure terminal
Router(config)# hostname R1
R1(config)# mpls ip
R1(config)# interface FastEthernet0/0
R1(config-if)# ip address 172.16.1.17 255.255.255.240
R1(config-if)# no shutdown
R1(config-if)# interface Serial0/0/0
R1(config-if)# ip address 192.168.10.1 255.255.255.252
R1(config-if)# clock rate 512000
R1(config-if)# bandwidth 512
R1(config-if)# no shutdown
R1(config-if)# mpls ip
R1(config-if)# interface Loopback0
R1(config-if)# ip address 1.1.1.1 255.255.255.255
R1(config-if)# end
```

Zde je vhodné poznamenat, že pomocí příkazu `clock rate` byla efektivně nastavena fyzická šířka pásma sériového spoje mezi směrovači R1 a R2 na 512 kb/s. Příkaz `bandwidth` a udaná hodnota je pouze logická metrika, která se využívá při různých výpočtech pro dané rozhraní. Dalším rozdílem mezi příkazy je to, že hodnotu `clock rate` zapisujeme v b/s, kdežto `bandwidth` v kb/s.

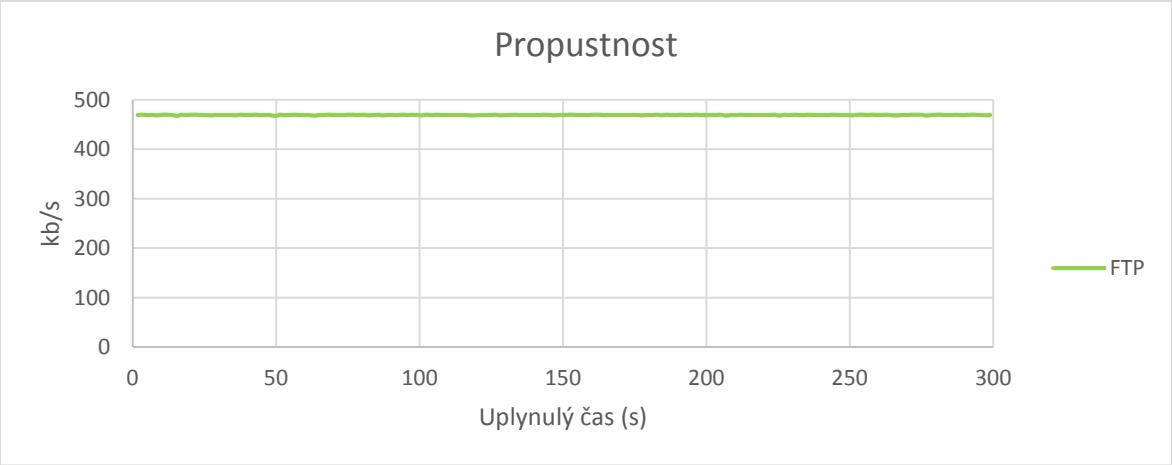
Jako směrovací protokol byl zvolen OSPF. Jedná se o tzv. protokol stavu linky, který umožňuje směrovačům si dynamicky vyměňovat informace o připojených sítích.

Následuje ukázka konfigurace OSPF na R1:

```
R1(config)# router ospf 1
R1(config-router)# network 1.1.1.1 0.0.0.0 area 0
R1(config-router)# network 172.16.1.16 0.0.0.15 area 0
R1(config-router)# network 192.168.10.0 0.0.0.3 area 0
```

4.1.1 Test sítě bez kvality služeb

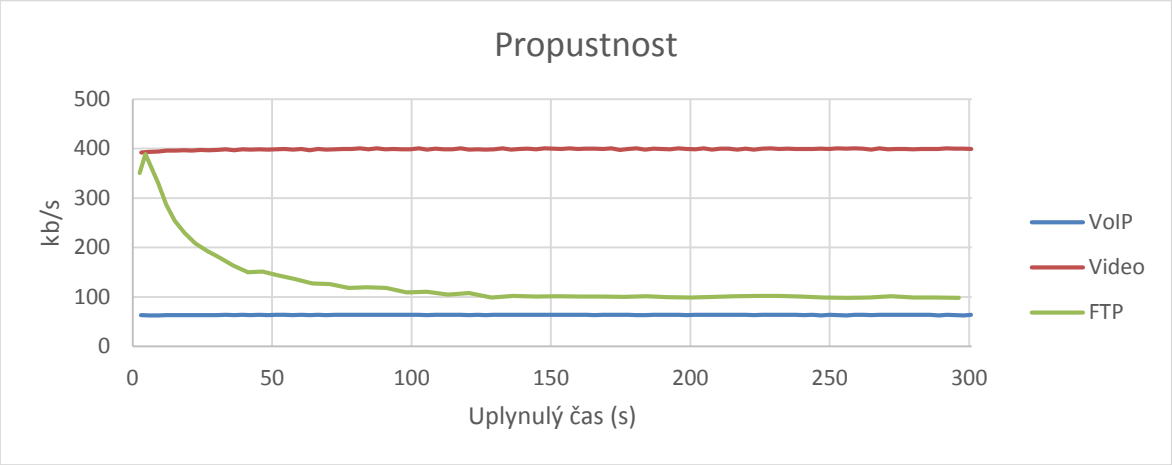
Pro ověření propustnosti (throughput) sítě byl proveden pomocí IxChariot test zátěže sítě provozem FTP. Zde je třeba si uvědomit rozdíl mezi šířkou pásma, což je teoretická maximální rychlost přenosu dat médiem v ideálních podmínkách, a propustností, což je rychlost přenosu dat skutečná a je typicky menší než šířka pásma. Zjednodušeně lze říci, že maximální teroreticky možná propustnost se rovná šířce pásma. Výsledek testu viz obrázek 17, průměrná propustnost dosahuje hodnoty 469,2 kb/s. Provést test nejprve pouze jedním datovým tokem je vhodné z toho důvodu, že při více generovaných datových tocích součet jejich propustností v programu IxChariot bude pravděpodobně vyšší než maximální dostupná šířka pásma. To je způsobeno tím, že „ne všechny páry endpoint odesílají data najednou, ale sumarizace časových záznamů je provedena jako kdyby tak činily" [26].



Obrázek 17 – Test propustnosti sítě, scénář č. 1

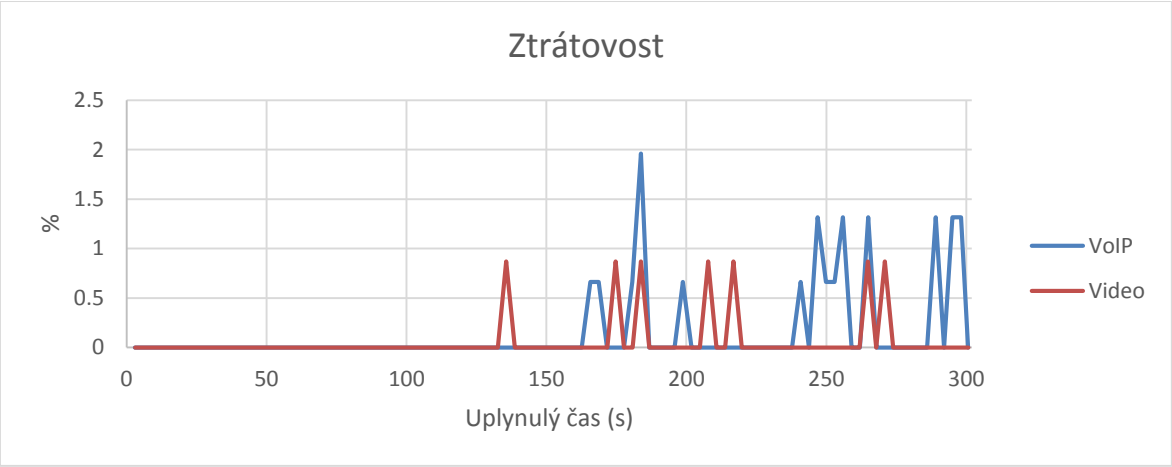
Dále byl proveden již plný test sítě, zatím bez jakýchkoliv mechanismů QoS. Při testu byly generovány 3 datové toky: FTP, video a hlasová komunikace VoIP. Přenosová rychlost (bitrate) pro video v kódování MPEG2 byla v programu IxChariot nastavena na 400 kb/s. Kodekem pro hlasový provoz byl G.711u (64 kb/s), který využívá beztrátové komprese a poskytuje nejlepší kvalitu hlasu.

Obrázek 18 zobrazuje propustnost a její změny v průběhu intervalu měření v testované síti.



Obrázek 18 – Propustnost bez QoS, scénář č. 1

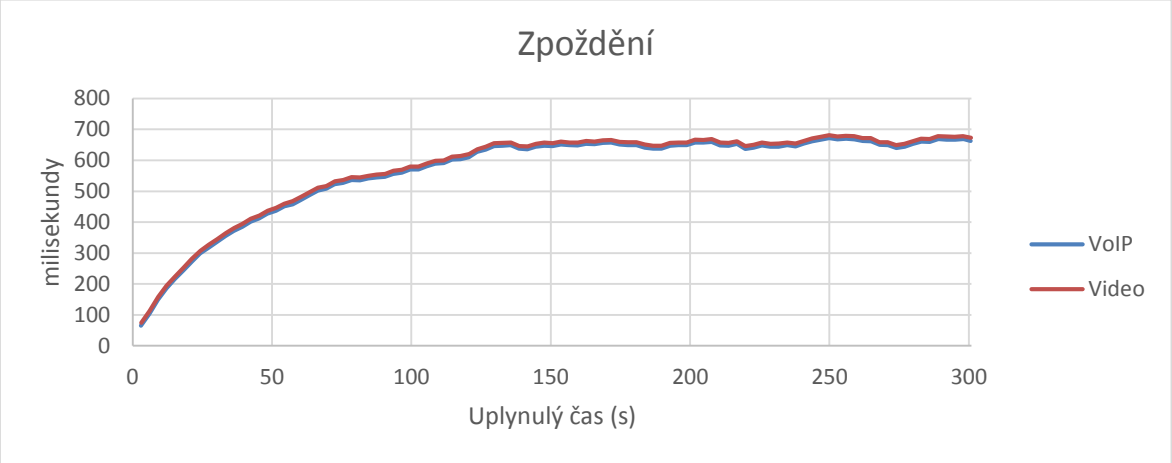
Jak patrno, video, využívající protokolu UDP, si dokáže „uzurpovat" šířku pásma před FTP. Narozdíl od toku FTP, který využívá protokol TCP zaručující spolehlivé doručování, však dochází k jisté ztrátovosti paketů (to platí i pro VoIP), více obrázek 19.



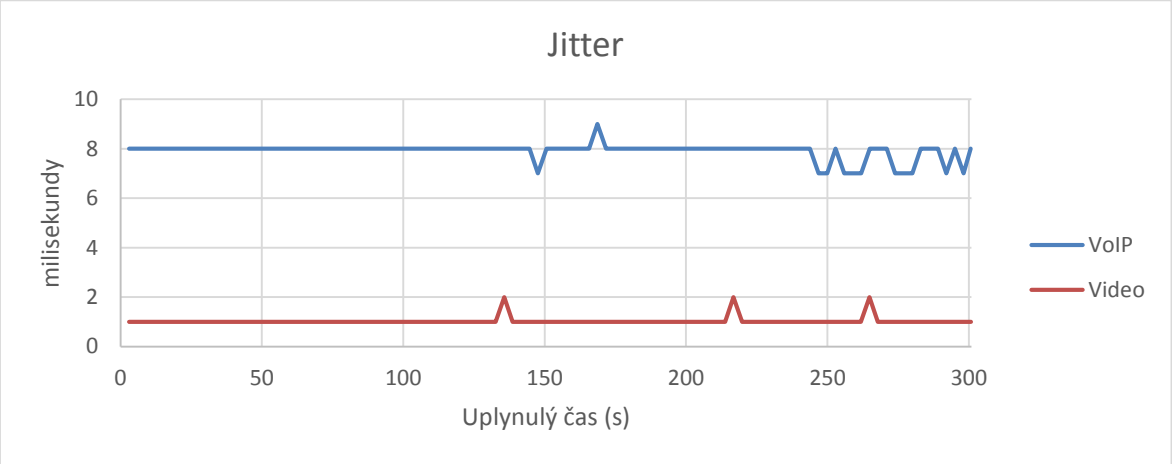
Obrázek 19 – Ztrátovost bez QoS, scénář č. 1

Dalšími sledovanými parametry jsou potom zpoždění (obrázek 20) a jitter (obrázek 21). Patrné jsou zejména vysoké hodnoty zpoždění a určité výkyvy v hodnotě jitteru. Doporučená hodnota zpoždění pro interaktivní aplikace typu video a VoIP je přitom <150 ms, ztrátovost paketů <1 % [27]. Síť však bez mechanismů QoS nebyla schopna těchto parametrů pro video a VoIP dosáhnout. Posledním ze sledovaných parametrů je tzv. odhad MOS (Mean Opinion Score) [28], kdy na škále od 1 do 5 je vyjádřena vnímaná

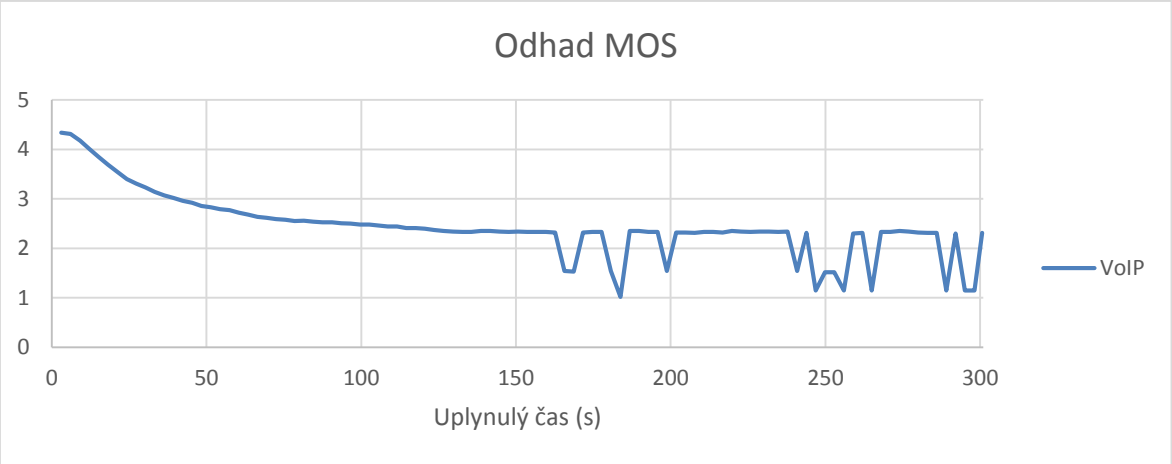
kvalita příjmu, kde 1 znamená znamená nejnižší kvalitu a 5 kvalitu nejvyšší. V testovací síti je kvalita hlasové komunikace bez QoS opravdu velmi slabá (obrázek 22) a vedla by ke značné nespokojenosti uživatele.



Obrázek 20 – Zpoždění bez QoS, scénář č. 1



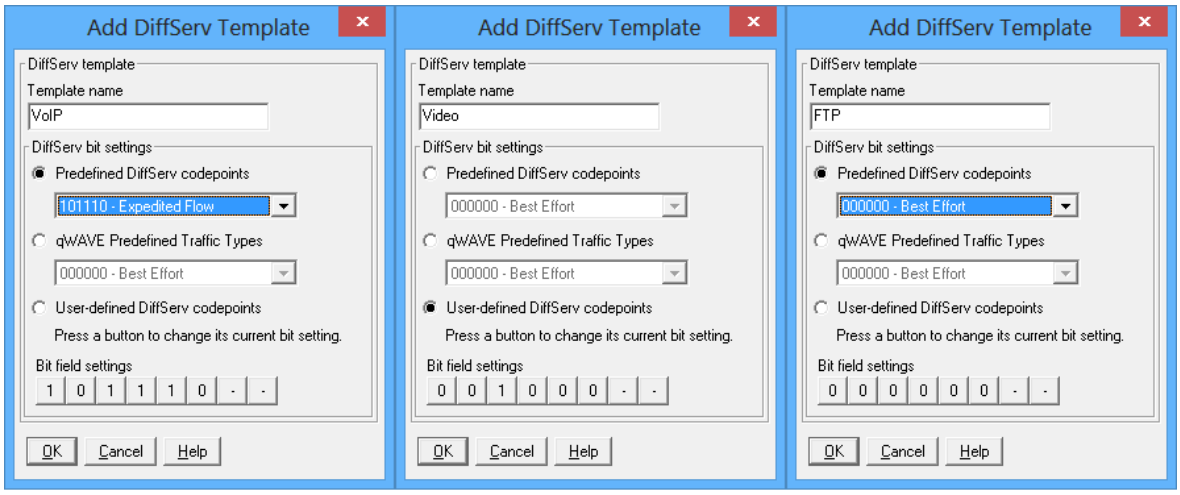
Obrázek 21 – Jitter bez QoS, scénář č. 1



Obrázek 22 – Odhad MOS bez QoS, scénář č. 1

4.1.2 Test sítě s kvalitou služeb

Následuje konfigurace mechanismů QoS. Program IxChariot umožňuje přímo nastavit hodnoty polí DSCP pro pakety jednotlivých toků. Pro VoIP bylo zvoleno Expedited Forwarding PHB (hodnota DSCP 46), pro video bylo definováno PHB vlastní (hodnota DSCP 8) a tok FTP byl ponechán jako Best Effort (hodnota DSCP 0), více obrázek 23. Na vstupu do domény MPLS byly tyto značky považovány za důvěryhodné.



Obrázek 23 – Nastavení hodnot polí DSCP, scénář č. 1

Je důležité si uvědomit, že tyto značky se nachází v hlavičkách protokolu IP. Hlavní výhodou MPLS ovšem je, že právě s těmito hlavičkami nemusí pracovat. Hodnoty DSCP jsou proto na vstupu do domény MPLS implicitně převáděny do pole EXP v návěstí MPLS. Jelikož toto pole je pouze 3 bity dlouhé, jsou převedeny pouze první tři bity pole DSCP. DSCP Expedited Forwarding se tak v poli EXP bude rovnat 5, hodnota EXP pro video bude 1 a pro FTP logicky 0. Na obrázku 24 je zkrácený výpis po použití příkazu `debug mpls packet` na směrovači R3. Prostřední hodnota v hranatých závorkách je právě zmíněnou hodnotou pole EXP.

```
*May 28 01:25:02.855: MPLS les: Se0/0/0: rx: Len 1364 Stack {19 1 127} - ipv4 da
ta
*May 28 01:25:02.855: MPLS les: Se0/0/0: rx: Len 208 Stack {19 5 127} - ipv4 dat
a
*May 28 01:25:02.855: MPLS les: Se0/0/0: rx: Len 208 Stack {19 5 127} - ipv4 dat
a
*May 28 01:25:03.627: MPLS les: Se0/0/0: rx: Len 208 Stack {19 5 127} - ipv4 dat
a
*May 28 01:25:03.639: MPLS les: Se0/0/0: rx: Len 208 Stack {19 5 127} - ipv4 dat
a
*May 28 01:25:03.639: MPLS les: Se0/0/1: rx: Len 60 Stack {18 0 127} - ipv4 data

*May 28 01:25:03.639: MPLS les: Se0/0/0: rx: Len 60 Stack {19 0 127} - ipv4 data

*May 28 01:25:03.643: MPLS les: Se0/0/1: rx: Len 60 Stack {18 0 127} - ipv4 data
```

Obrázek 24 – Debug mpls packet

V případě této testovací sítě má smysl konfiguraci QoS provést pouze na rozhraní S0/0/0 směrovače R1. Nejen proto, že vnitřní oblast domény by se již měla zabývat pouze

přepojováním paketů, ale v doméně MPLS funguje tzv. Penultimate Hop Popping (PHP), kdy před předáním paketu na LER (zde konkrétně R2) na okraji domény je odstraněno návěstí MPLS a nelze tak již provádět politiku QoS založenou na hodnotě pole EXP, neboť odstranění je provedeno ještě před tím, než je možné ji uplatnit.

Při konfiguraci QoS je neprve třeba vytvořit jednotlivé třídy provozu na základě zvolených pravidel.

```
R1(config)# class-map match-all VOICE
R1(config-cmap)# match mpls experimental topmost 5
R1(config-cmap)# exit
R1(config)# class-map match-all VIDEO
R1(config-cmap)# match mpls experimental topmost 1
R1(config-cmap)# exit
```

Následně je třeba nastavit tzv. mapu politik (policy-map), kde budou určena pravidla zacházení s jednotlivými třídami.

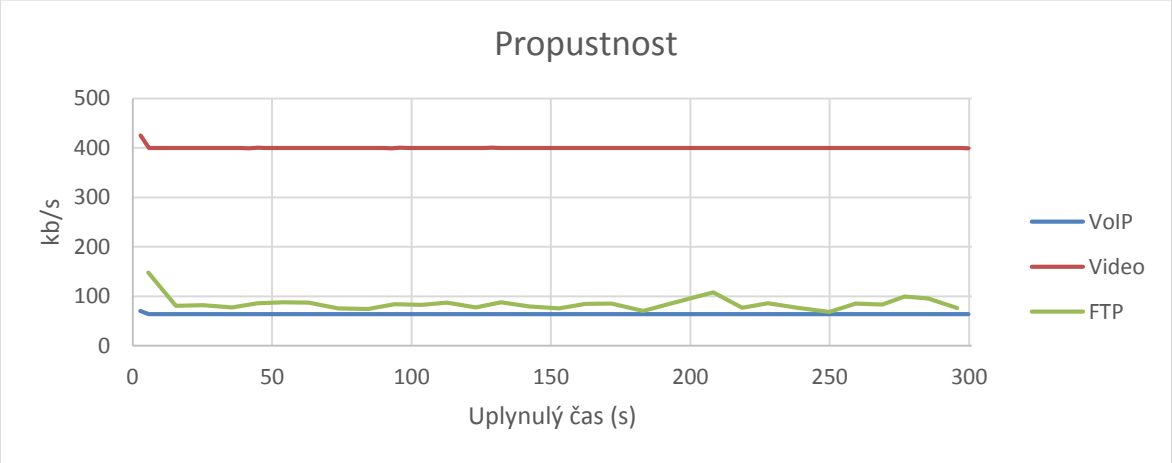
```
R1(config)# policy-map POLITIKA
R1(config-pmap)# class VOICE
R1(config-pmap-c)# priority percent 15
R1(config-pmap-c)# exit
R1(config-pmap)# class VIDEO
R1(config-pmap-c)# priority percent 40
R1(config-pmap-c)# exit
R1(config-pmap)# class class-default
R1(config-pmap-c)# fair-queue
R1(config-pmap-c)# queue-limit 10
R1(config-pmap-c)# random-detect
R1(config-pmap-c)# exit
```

Příkaz `priority` vyhrazuje dané třídě minimální šířku pásma v případě zahlcené sítě v procentech z celkové šířky pásma. V případě zahlcení je také třída omezována (policing), aby vyhrazenou šířku pásma nepřekračovala. Jedná se vlastně o Low Latency Queuing, kdy třída je umístěna do prioritní fronty uvnitř CBWFQ. Tím je také pro třídu redukováno zpoždění a jitter, což je obzvlášť vhodné pro hlasový provoz a také například video. Do třídy `class-default` spadá veškerý síťový provoz, který nepatří do některé z námi definovaných tříd. Pro tuto třídu byla nastavena fronta WFQ, která by měla spravedlivě dělit šířku pásma mezi jednotlivé toky (v tomto scénáři se však bude jednat pouze o tok jeden, FTP) pomocí příkazu `fair-queue` a limit množství paketů ve frontě pomocí `queue-limit` na 10. Po použití příkazu `random-detect` se fronta bude inteligentním odhazováním paketů snažit zabránit svému zaplnění. To má však význam pouze pro datové toky postavené nad TCP, které jsou schopné zpomalit vysílání v reakci na ztrátu (odhození) paketů.

Mapu politik je poté třeba aplikovat na konkrétní odchozí rozhraní ve zvoleném směru.

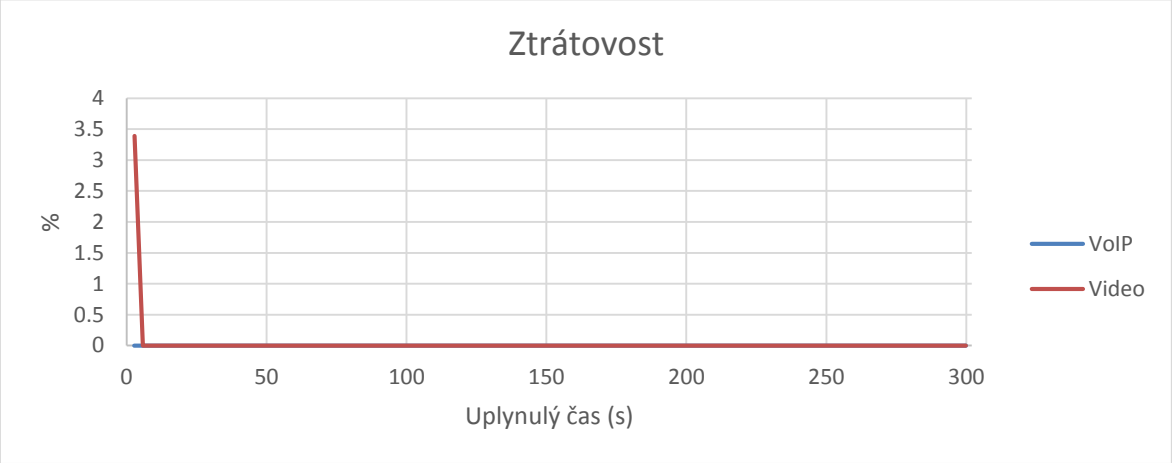
```
R1(config)# interface Serial0/0/0
R1(config-if)# service-policy output POLITIKA
```

Následuje provedení testu sítě s QoS simulovanou zátěží, nastavení jednotlivých toků zůstává stejné jako při testu bez QoS. Obrázek 25 ukazuje propustnost jednotlivých toků, propustnost videa dle předpokladu na úkor FTP stoupla.



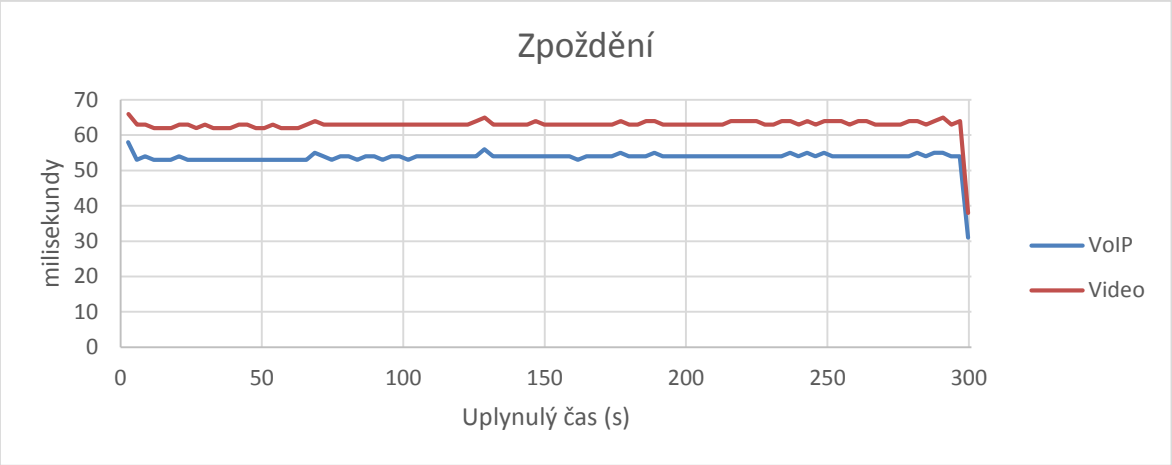
Obrázek 25 – Propustnost s QoS, scénář č. 1

Ztrátovost paketů pro video a VoIP klesla k nule (vice obrázek 26).



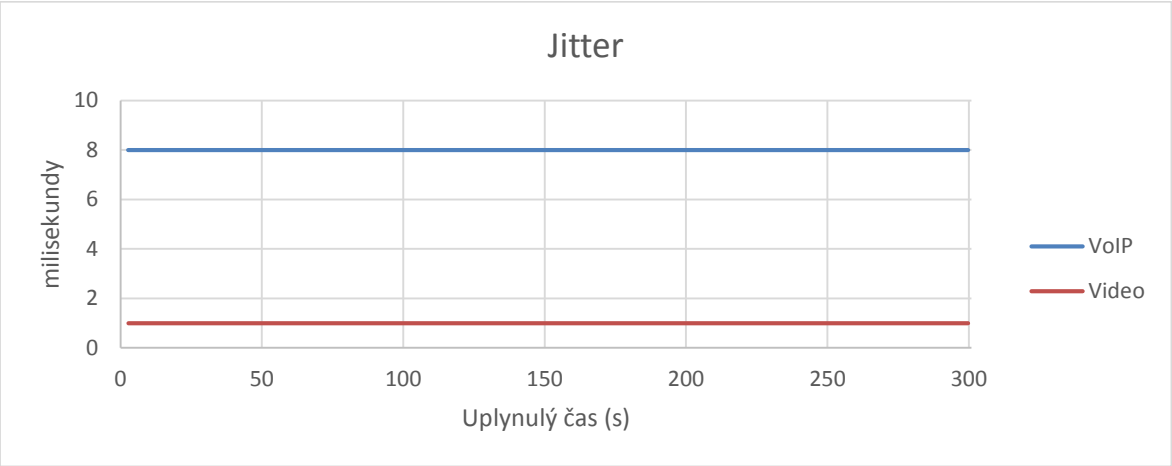
Obrázek 26 – Ztrátovost s QoS, scénář č. 1

Nasazením LLQ pro video a VoIP bylo dosaženo až desetinásobného zlepšení hodnot zpoždění díky umístění tříd v prioritní frontě (obrázek 27). Významný pokles hodnot zpoždění před koncem měření je způsoben tím, že program několik sekund před ukončením měření již přestal generovat tok FTP.



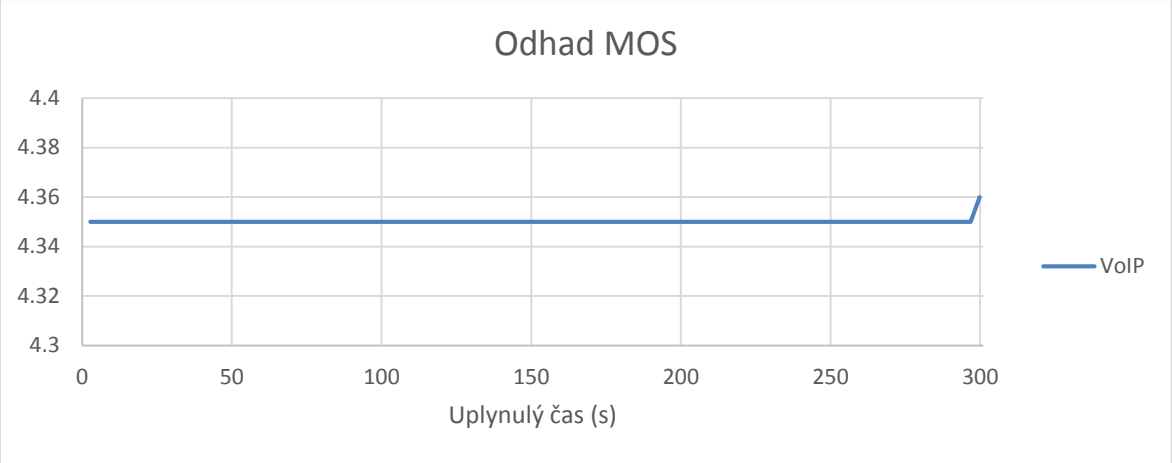
Obrázek 27 – Zpoždění s QoS, scénář č. 1

Jitter také dosahuje vynikajících a stabilních hodnot (obrázek 28).



Obrázek 28 – Jitter s QoS, scénář č. 1

U odhadu MOS došlo také k velmi výraznému zlepšení, komunikace VoIP by byla nyní uživatelem hodnocena jako kvalitní a bezproblémová (obrázek 29).



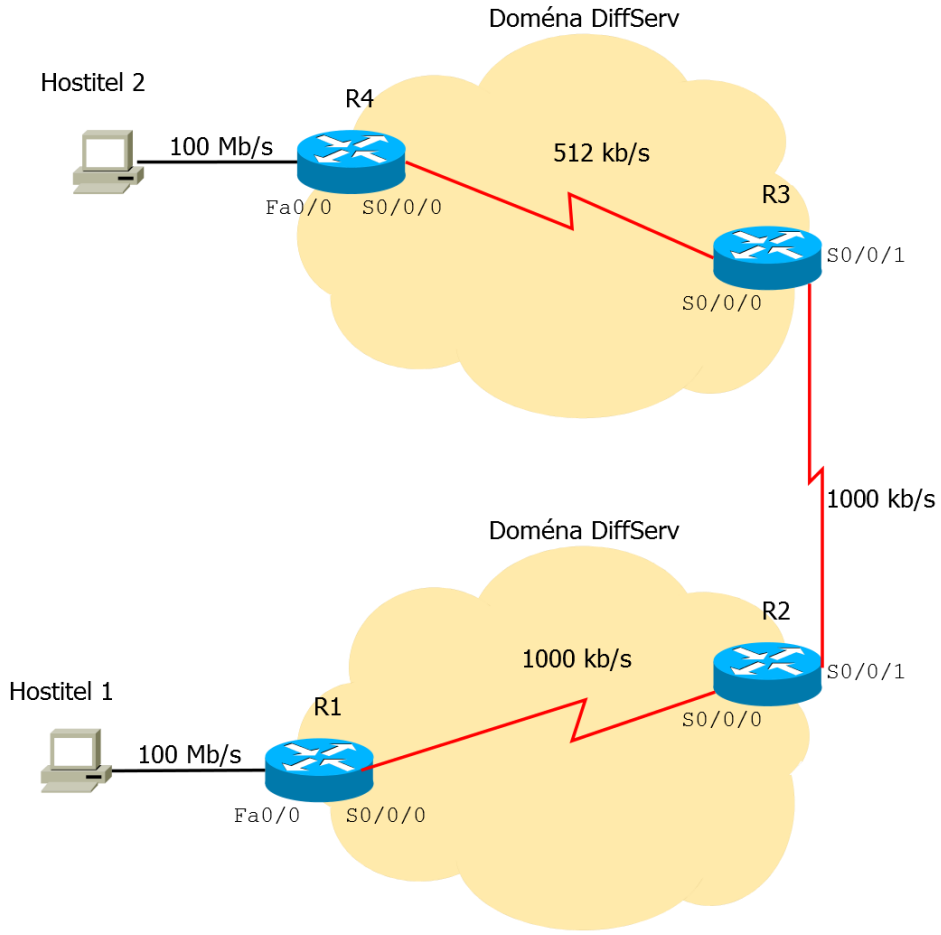
Obrázek 29 – Odhad MOS s QoS, scénář č. 1

4.1.3 Shrnutí výsledků scénáře č. 1

Tabulka v příloze B.1 bakalářské práce přehledně shrnuje dosažené parametry pro jednotlivé toky před a po implementaci QoS. Při porovnání je patrné, že u videa a provozu VoIP bylo dosaženo výrazného zlepšení všech sledovaných parametrů při implementaci QoS pomocí LLQ. Např. průměrné zpoždění pro VoIP kleslo z 566 ms na 54 ms, což představuje zlepšení o cca 1048 %. LLQ je tedy skutečně silným nástrojem QoS. Preference videa a VoIP nicméně v důsledku měla nepříznivý vliv na propustnost FTP jakožto neprioritního toku zpracovaného metodou Best Effort, kde propustnost klesla z průměrných 124,4 kb/s na 84 kb/s.

4.2 Kvalita služby a rozhraní domén diferencovaných služeb

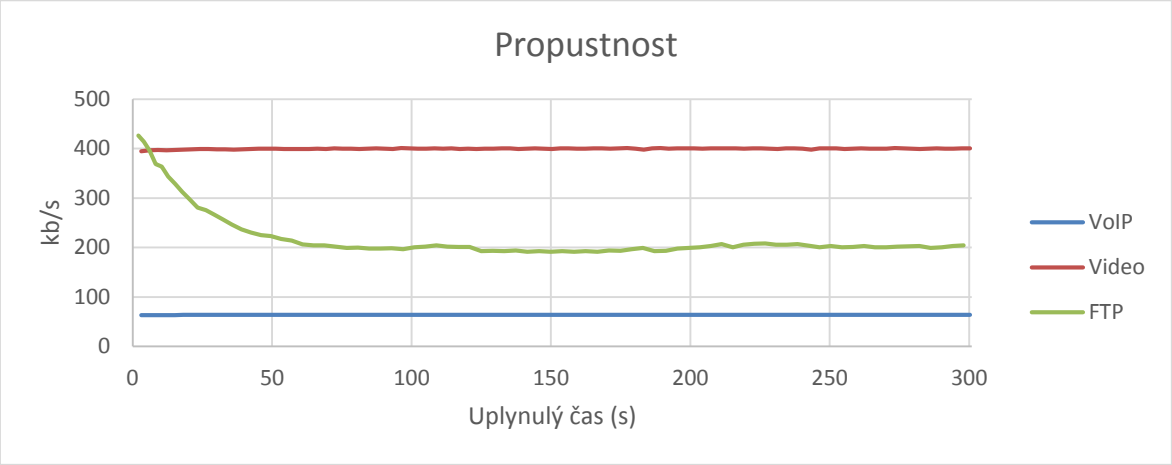
V druhém scénáři bude implementace QoS provedena v síti s dvěma doménami DiffServ. Obrázek 30 představuje podobu testovací topologie. Podrobné nastavení adresace IP sítě je potom k dispozici v příloze A.2 bakalářské práce. Síťový provoz bude opět generován na hostiteli 1 a poté směrován od hostitele 1 k hostiteli 2. Jako směrovací protokol byl zvolen OSPF.



Obrázek 30 – Topologie scénáře č. 2

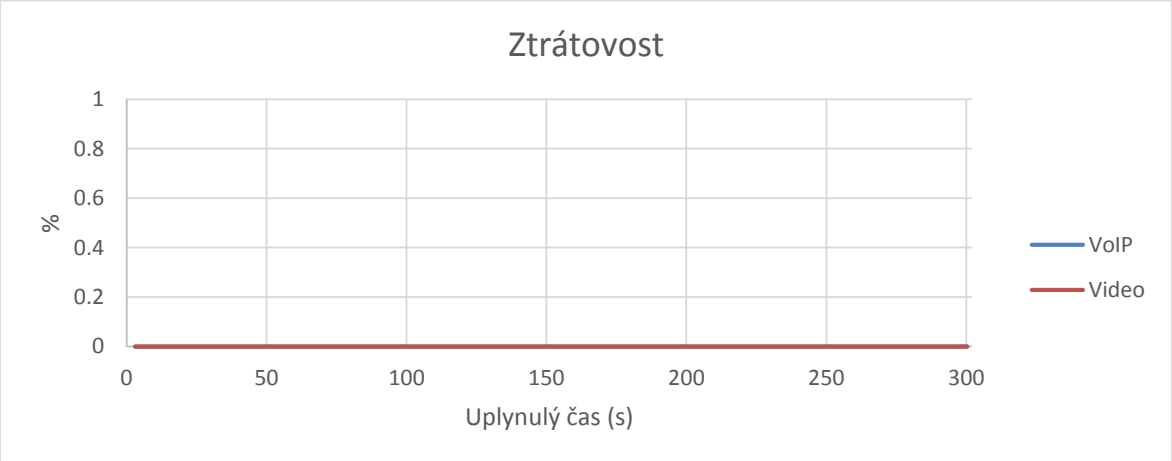
4.2.1 Test sítě bez kvality služeb

Nejdříve byl proveden test sítě bez mechanismů QoS. Při testu byly generovány opět 3 datové toky: FTP, video a hlasová komunikace VoIP. Přenosová rychlost pro video v kódování MPEG2 byla v programu IxChariot nastavena na 400 kb/s. Kodekem pro hlasový provoz byl G.711u (64 kb/s). První test se zaměřil na propustnost datových toků, názorné vyobrazení propustnosti a její změny v průběhu intervalu měření v testované síti při simulované zátěži viz obrázek 31. Průběh chování sítě nevykazuje zvláštní odlišnosti od scénáře č. 1, tok FTP však v porovnání získává až dvojnásobnou propustnost.



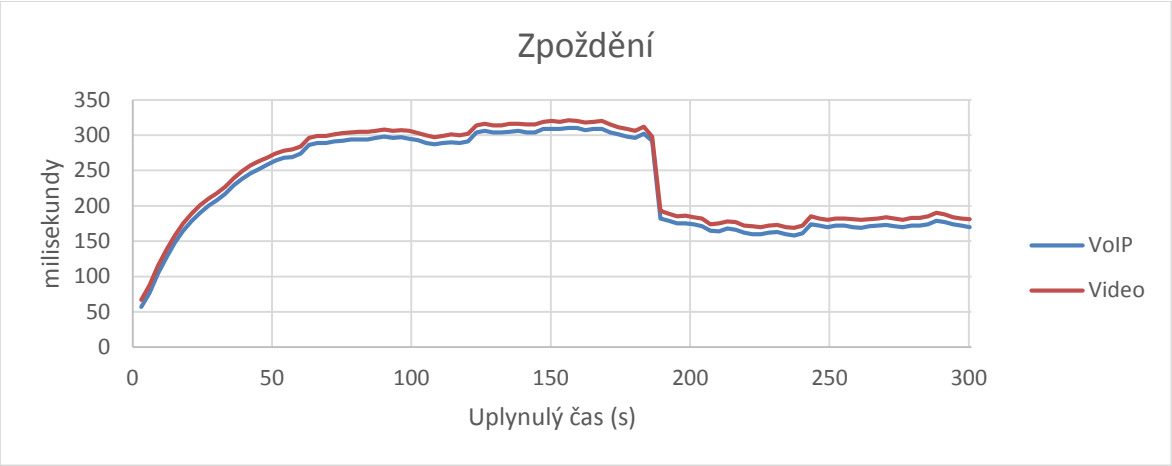
Obrázek 31 – Propustnost bez QoS, scénář č. 2

Dále byla sledována ztrátovost paketů (obrázek 32). Všechny pakety byly v testovací síti doručeny a ztrátovost tak byla nulová.

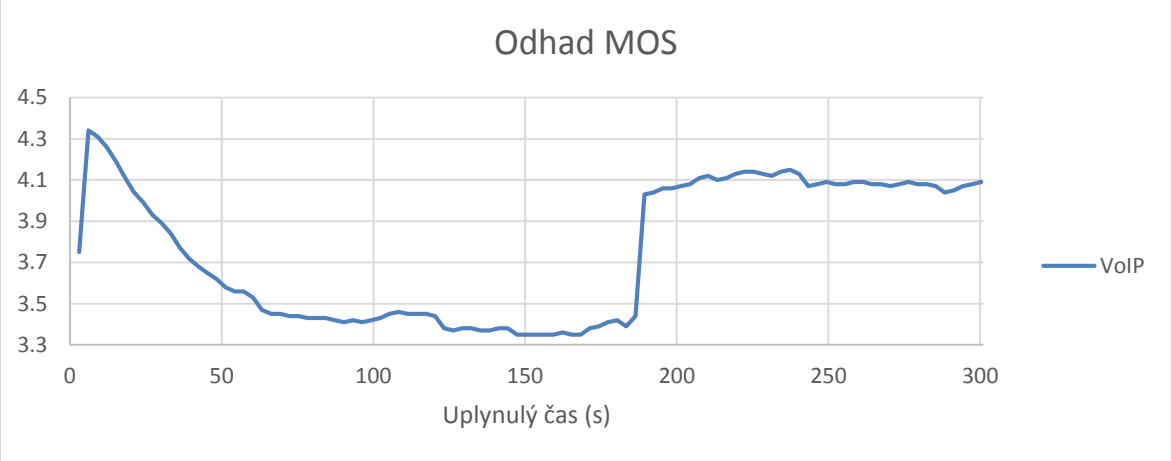


Obrázek 32 – Ztrátovost bez QoS, scénář č. 2

Ke sledovaným parametrům patří také zpoždění. Hodnoty pro VoIP opět překračovaly doporučených 150 ms, více obrázek 33. Za povšimnutí stojí propad hodnot zpoždění mezi 150 s a 200 s času měření. Ten se vzápětí projevil jako zlepšení odhadu MOS (obrázek 34), kde hodnoty MOS nižší než 3,5 by již pravděpodobně obtěžovaly uživatele.

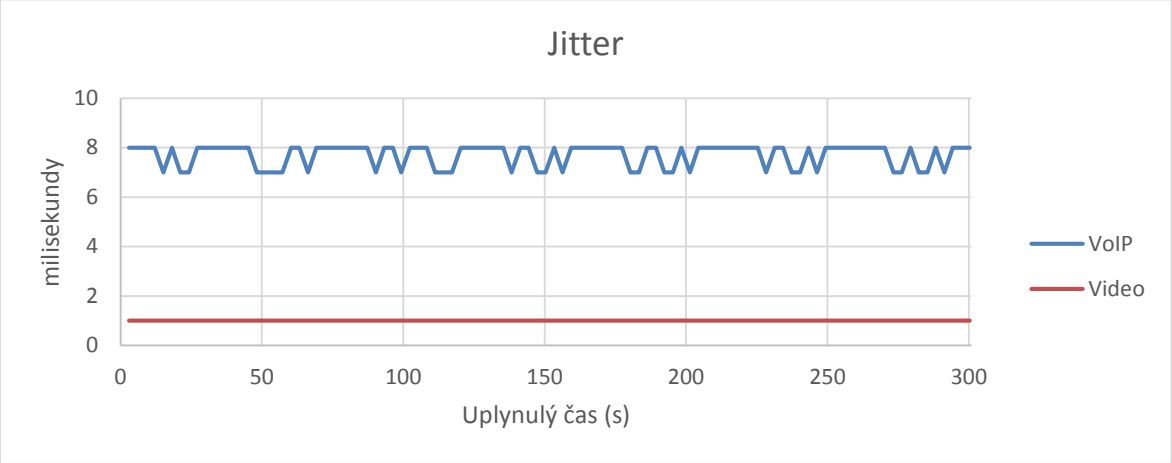


Obrázek 33 – Zpoždění bez QoS, scénář č. 2



Obrázek 34 – Odhad MOS bez QoS, scénář č. 2

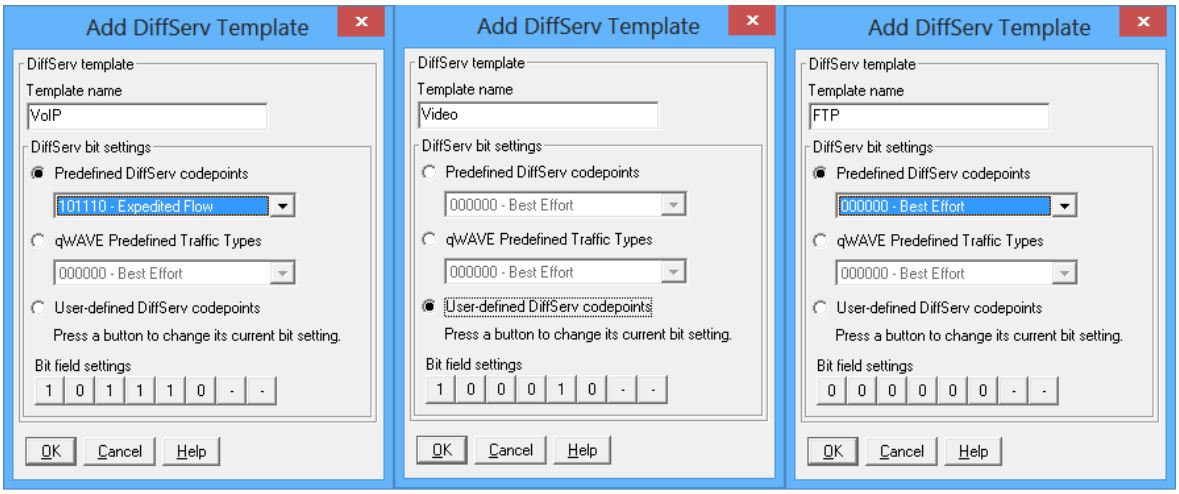
Posledním z parametrů je jitter, kde lze u hlasové komunikace vysledovat jisté výkyvy v jeho hodnotách, viz obrázek 35.



Obrázek 35 – Jitter bez QoS, scénář č. 2

4.2.2 Test sítě s kvalitou služeb

Následuje konfigurace mechanismů QoS. Programem IxChariot byly opět přímo nastaveny hodnoty polí DSCP pro pakety jednotlivých toků. Pro VoIP bylo zvoleno Expedited Forwarding PHB (hodnota DSCP 46), pro video jako Assured Forwarding PHB 41 (hodnota DSCP 34) a tok FTP byl ponechán jako Best Effort (hodnota DSCP 0), viz obrázek 36.



Obrázek 36 – Nastavení hodnot polí DSCP, scénář č. 2

Síťový provoz přichází do domény přilehlé k hostiteli 1 s již označeným polem DSCP a toto značení je považováno za důvěryhodné. Avšak značení provozu, který z této domény přichází do domény DiffServ přilehlé k hostiteli 2, již za plně důvěryhodné považováno není – v reálné situaci může např. každá doména podléhat jinému správci. Domény se navíc liší i tím, že uvnitř používají spoje s odlišnou kapacitou (1000 kb/s vs. 512 kb/s). Směrovač R3 proto příchozí provoz přeznačí pro potřeby domény DiffServ, na jejímž okraji leží. PHB generovaného toku videa v první doméně odpovídá AF 41 PHB, tedy

nejvyšší třídě Assured Forwarding. AF 41 PHB je na směrovači R3 přeznačeno na AF 31 PHB (hodnota DSCP 26), tedy nižší třídu Assured Forwarding. Dle doporučení by taková třída měla mít k dispozici menší šířku pásma než třída vyšší [17]. Ostatní značení je ponecháno. Příloha C.1 bakalářské práce obsahuje porovnání značení paketu toku videa odchozího z hostitele 1 (tedy před přeznačením) a již přeznačeného paketu příchozího na hostitele 2.

Konfigurace tříd provozu na směrovači R1:

```
R1(config)# class-map match-all VOICE
R1(config-cmap)#  match dscp ef
R1(config-cmap)# exit
R1(config)# class-map match-all VIDEO
R1(config-cmap)# match dscp af41
R1(config-cmap)# exit
```

Nastavení mapy politik `POLITIKA`:

```
R1(config)# policy-map POLITIKA
R1(config-pmap)# class VOICE
R1(config-pmap-c)# priority percent 8
R1(config-pmap-c)# exit
R1(config-pmap)# class VIDEO
R1(config-pmap-c)# bandwidth percent 40
R1(config-pmap-c)# exit
R1(config-pmap)# class class-default
R1(config-pmap-c)# fair-queue
R1(config-pmap-c)# queue-limit 10
R1(config-pmap-c)# random-detect
R1(config-pmap-c)# exit
```

VoIP tak v důsledku získává 8 procent z celkové šířky pásma 1000 kb/s v prioritní frontě LLQ. Pro video bylo pomocí příkazu `bandwidth` rezervováno minimálně 40 % celkové šířky pásma v případě zahlcení sítě. `Bandwidth` je implementován pomocí CBWFQ, neprovádí omezování provozu a neposkytuje umístění v prioritní frontě. Nastavení pro třídu `class-default` je stejné jako v případě scénáře č. 1.

Aplikace mapy politik `POLITIKA` na konkrétní odchozí rozhraní:

```
R1(config)# interface Serial0/0/0
R1(config-if)# service-policy output POLITIKA
```

Konfigurace mapy tříd na směrovači R3:

```
R3(config)# class-map match-all VOICE
R3(config-cmap)# match dscp ef
R3(config-cmap)# exit
R3(config)# class-map match-all VIDEO_IN
R3(config-cmap)# match dscp af41
R3(config)# class-map match-all VIDEO_OUT
R3(config-cmap)# match dscp af31
```

Pro tok videa byly nastaveny 2 třídy, neboť se na směrovači R3 budou uplatňovat dvě různé politiky, jedna pro provoz, který bude přeznačen před vstupem do domény DiffServ, a druhá pro provoz odchozí ze směrovače R3 do vnitřní oblasti domény DiffServ.

Konfigurace mapy politik `POLITIKA` na směrovači R3:

```
R3(config)# policy-map POLITIKA
R3(config-pmap)# class VOICE
R3(config-pmap-c)# priority percent 15
R3(config-pmap-c)# exit
R3(config-pmap)# class VIDEO_OUT
R3(config-pmap-c)# bandwidth percent 40
R3(config-pmap-c)# police 250000 conform-action transmit
exceed-action drop
R3(config-pmap-c-police)# exit
R3(config-pmap)# class class-default
R3(config-pmap-c)# fair-queue
R3(config-pmap-c)# queue-limit 10
R3(config-pmap-c)# random-detect
R3(config-pmap-c)# exit
```

Pro hlasový provoz VoIP bylo vyhrazeno 15 % z celkové šířky pásma 512 kb/s v prioritní frontě LLQ. Pro video bylo rezervováno minimálně 40 % celkové šířky pásma v případě zahlcení sítě. Tok videa bude nyní navíc omezován, pomocí příkazu `police`. Při překročení přenosové rychlosti 250000 b/s budou pakety toku odhazovány zahazovačem. Nastavení pro třídu `class-default` se nadále nemění.

Aplikace mapy politik `POLITIKA` na konkrétní odchozí rozhraní:

```
R3(config)# interface Serial0/0/0
R3(config-if)# service-policy output POLITIKA
```

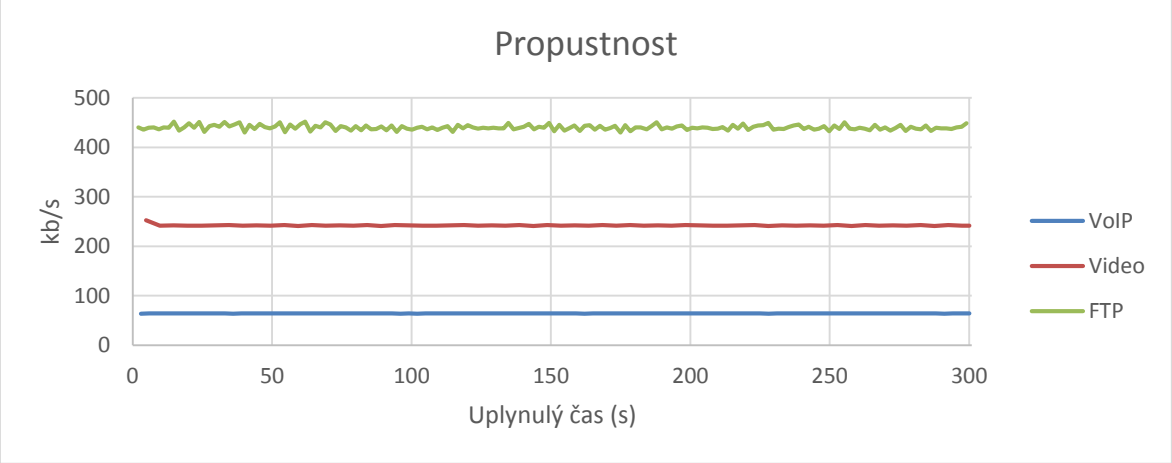
Konfigurace mapy politik `PREZNACENI` pro přeznačování paketů na směrovači R3 (pro vstupující pakety třídy `VIDEO_IN` vždy dojde ke změně hodnoty pole DSCP na 26) :

```
R3(config)# policy-map PREZNACENI
R3(config-pmap)# class VIDEO_IN
R3(config-pmap-c)# set dscp af31
R3(config-pmap-c)# exit
```

Aplikace mapy politik `PREZNACENI` na konkrétní vstupní rozhraní:

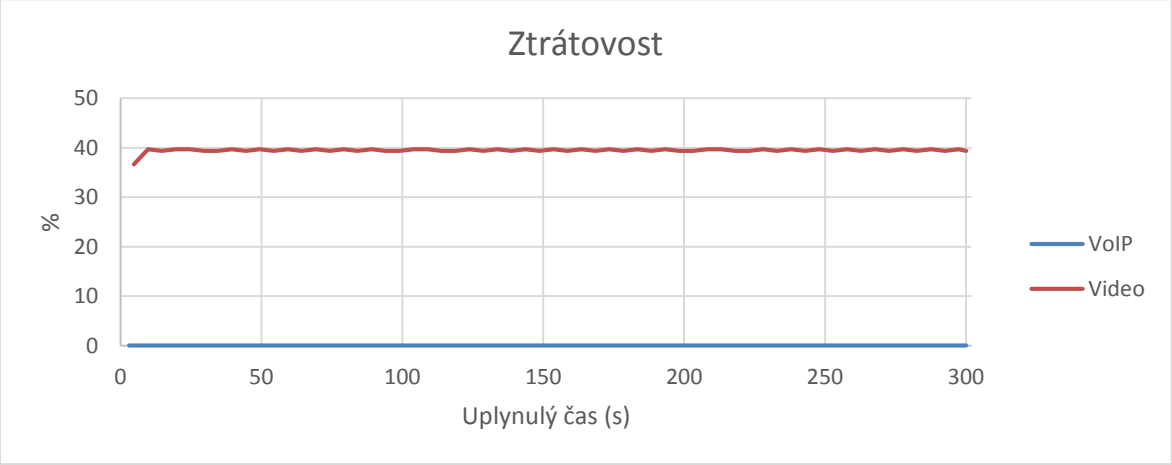
```
R3(config)# interface Serial0/0/1
R3(config-if)# service-policy input PREZNACENI
```

Následovalo provedení testu sítě s QoS simulovanou zátěží, nastavení jednotlivých toků zůstává stejné jako při testu bez QoS. Obrázek 37 ukazuje propustnost jednotlivých toků, kdy na propustnosti toku videa se jasně projevuje efekt práce zahazovače a tok FTP získává větší propustnost v porovnání se sítí bez implementace QoS.



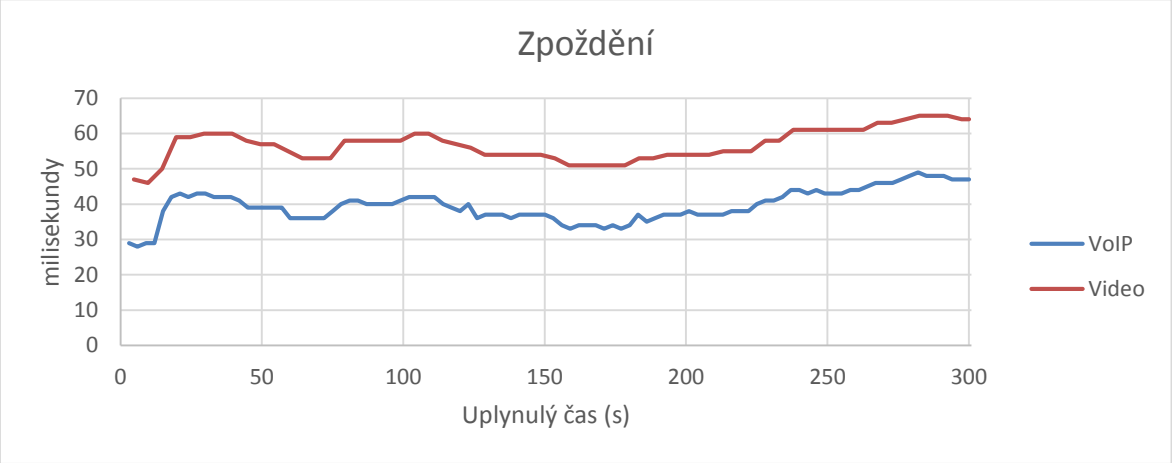
Obrázek 37 – Propustnost s QoS, scénář č. 2

Dále byla sledována ztrátovost paketů. Jelikož přenosová rychlost videa byla u zdroje nastavena na 400 kb/s a je tak vyšší než hranice pro omezování toku (250 kb/s), činnost zahazovače se jasně projevuje ve ztrátovosti paketů toku videa v porovnání s případem, kdy není v síti aktivní omezování provozu. Zahazovač při překročení definovaného profilu toku začne odhazovat pakety a v důsledku tak musí dojít ke zvýšení ztrátovosti, viz obrázek 38.



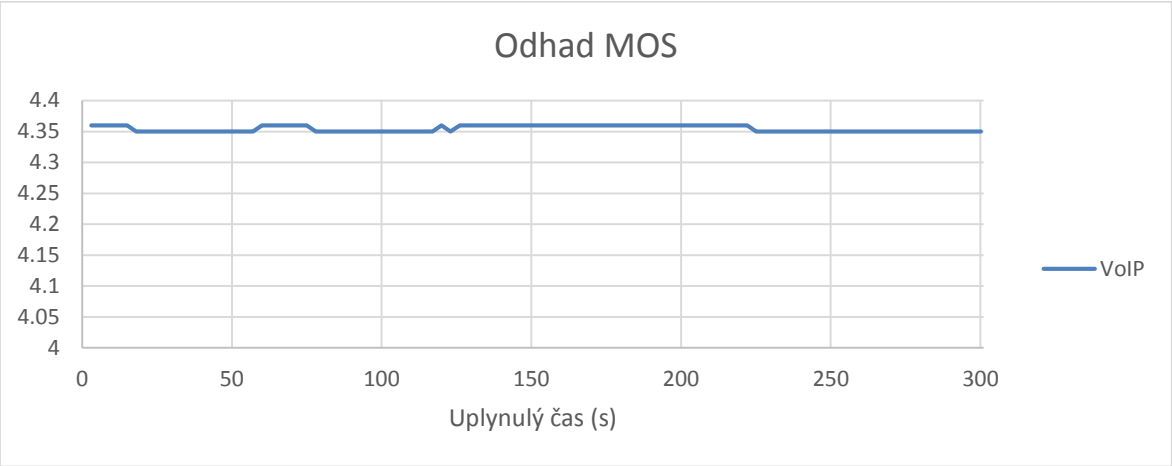
Obrázek 38 – Ztrátovost s QoS, scénář č. 2

Z obrázku 39 je patrné, že pomocí mechanismů QoS bylo pro toky videa a VoIP dosaženo zlepšení hodnot zpoždění. Zejména u VoIP se jedná o vynikající hodnoty.



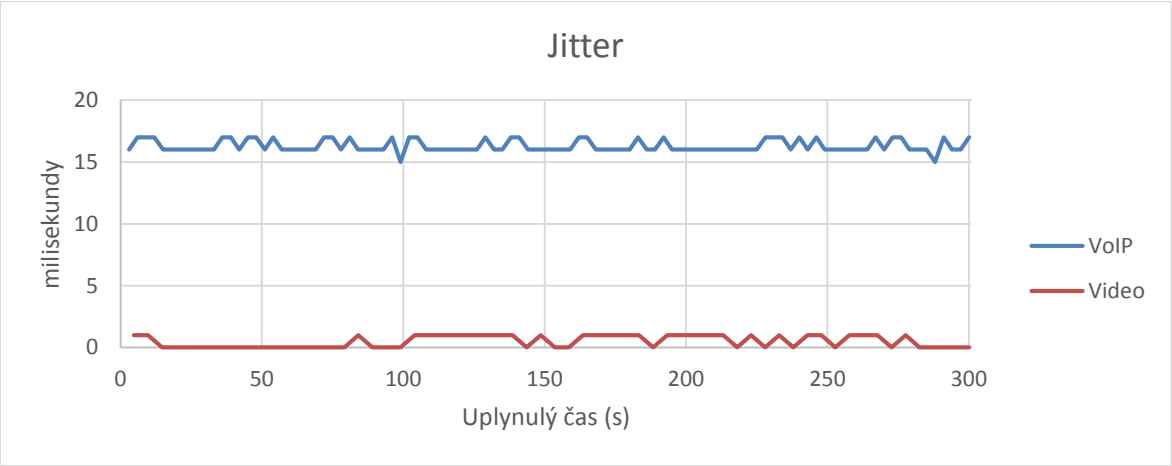
Obrázek 39 – Zpoždění s QoS, scénář č. 2

Nízké hodnoty zpoždění se pak (spolu s dalšími parametry) promítlo do odhadu MOS, kdy byla po celou dobu měření udržena vynikající kvalita komunikace VoIP. Více na obrázku 40.



Obrázek 40 – Odhad MOS s QoS, scénář č. 2

Posledním ze sledovaných parametrů byl jitter (obrázek 41). Poměrně překvapivým faktem bylo obecné zvýšení hodnot jitteru pro komunikaci VoIP. Jako možné vysvětlení se nabízí to, že intenzivní práce zahazovače nad tokem videa, kde došlo k odhození 39,4 % ze všech paketů toku, s sebou nese jistou režii a potenciálně vnáší do sítě vyšší jitter pro tok, nad kterým zahazovač nepracuje.



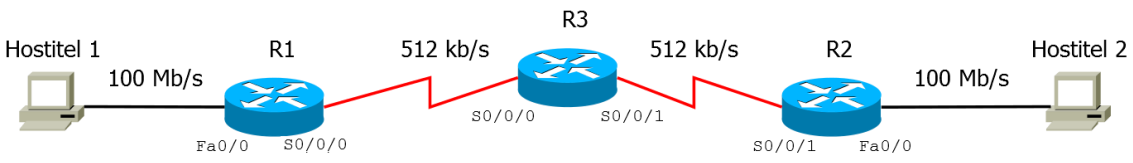
Obrázek 41 – Jitter s QoS, scénář č. 2

4.2.3 Shrnutí výsledků scénáře č. 2

Tabulka v příloze B.2 bakalářské práce přehledně shrnuje dosažené parametry pro jednotlivé toky před a po implementaci QoS. I druhý scénář potvrdil potřebu QoS pro zahlcenou síť, kdy opět došlo ke zlepšení sledovaných parametrů (kromě jitteru) a vyšší kvalitě komunikace VoIP. Zvýšený jitter komunikace VoIP však měl jen malý vliv na odhad MOS, kde bylo stále dosaženo vynikající kvality hovoru, a lze jej chápat jako jistou daň za přizpůsobení přenosové rychlosti toku videa potřebám konkrétní domény DiffServ pomocí omezování. To se v testu projevilo jako skutečně silný a kvalitní nástroj pro úpravu síťového provozu. Na rozhraní dvou testovacích domén bylo navíc provedeno přeznačování pole DSCP přeznačovačem, jehož korektní funkce byla potvrzena výstupem síťového analyzátoru, sledujícího síťový provoz odchozí z hostitele 1 a síťový provoz příchozí na hostitele 2.

4.3 AutoQoS

Ve třetím scénáři praktické části práce bude implementace QoS pomocí nástroje AutoQoS provedena v síti, jejíž topologie je představena na obrázku 42.



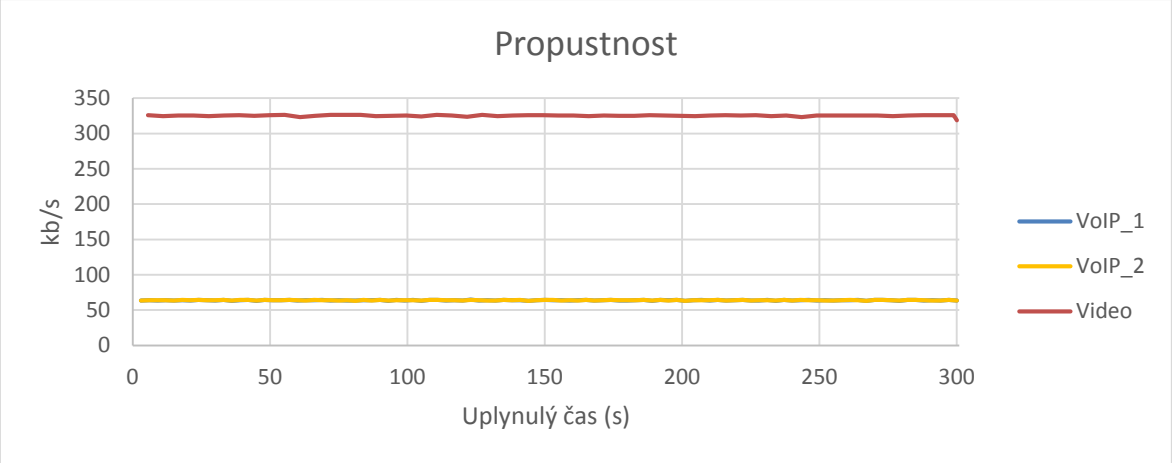
Obrázek 42 – Topologie scénaře č. 3

Síťový provoz byl generován na hostiteli 1 a poté od něj směrován k hostiteli 2. Podrobné nastavení adresace IP sítě je k dispozici v příloze A.1 bakalářské práce. Jako směrovací protokol byl zvolen OSPF.

AutoQos je způsob, jak na směrovačích a přepínačích Cisco jednoduše implementovat QoS pro komunikaci VoIP bez zvláštních znalostí a je tak atraktivní možností pro rychlé a spolehlivé zajištění kvality komunikace VoIP např. v malé podnikové síti, kde neflexibilita (s veškerým provoz kromě VoIP bude zacházeno metodou Best Effort) tohoto nástroje nebude překážkou. Síťový provoz je automaticky klasifikován, automaticky je také generována mapa tříd a mapa politik, což činí celou konfiguraci triviální. Existuje i komplexnější verze AutoQoS Enterprise, která své funkce neomezuje pouze na VoIP.

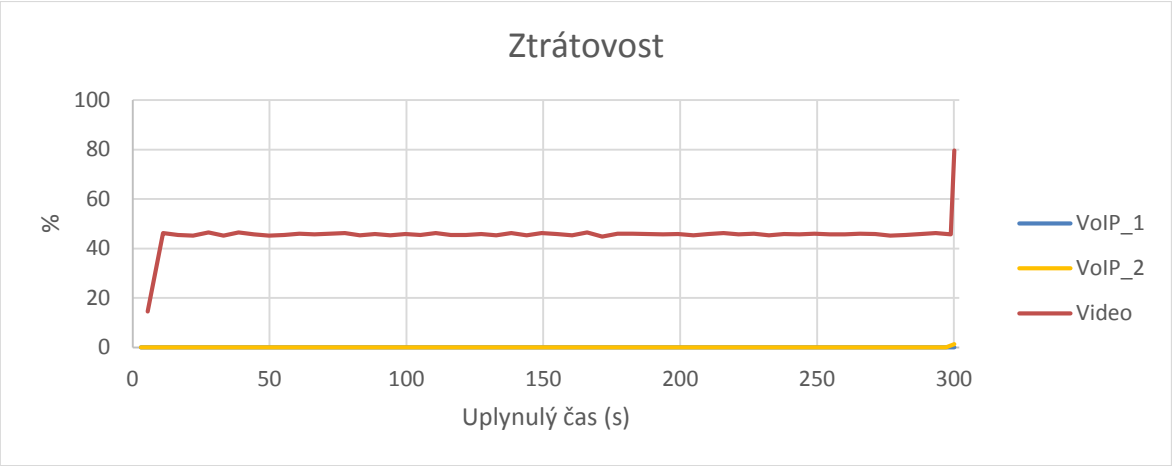
4.3.1 Test sítě bez kvality služeb

Nejdříve byl proveden test sítě bez mechanismů AutoQoS. Při testu byly generovány 3 datové toky: video a dva toky hlasové komunikace VoIP. Přenosová rychlost pro video v kódování MPEG2 byla v programu IxChariot nastavena na 600 kb/s. Kodekem pro hlasový provoz byl G.711u (64 kb/s). Názorné vyobrazení propustnosti a její změny v průběhu intervalu měření v testované síti při simulované zátěží viz obrázek 43.



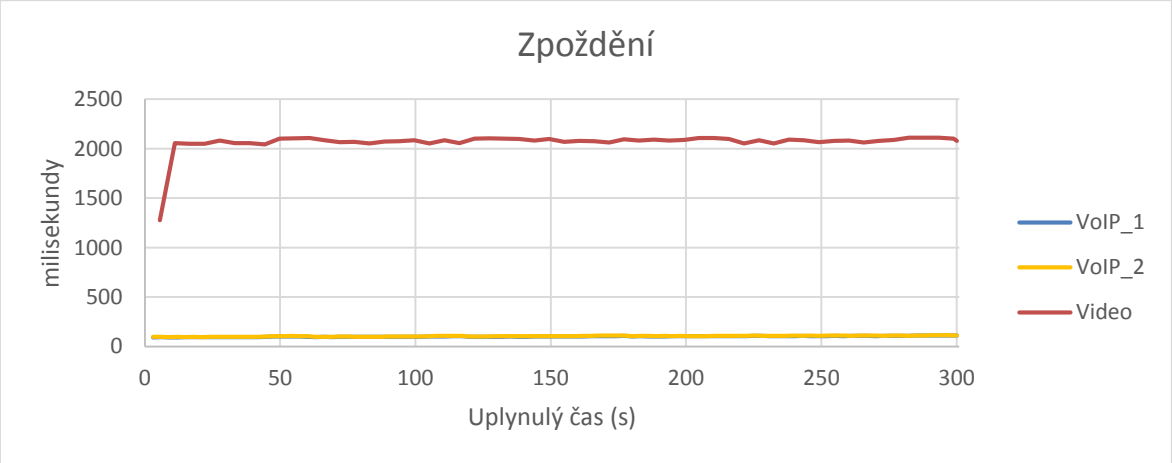
Obrázek 43 – Propustnost bez QoS, scénář č. 3

Dalším ze sledovaných parametrů byla ztrátovost. Na toku videa se projevuje fakt, že nastavená přenosová rychlost toku videa je vyšší než šířka pásma a nutně tak dochází ke ztrátám paketů, neboť protokol UDP není na rozdíl od TCP schopen přizpůsobit rychlost toku a postrádá schopnost spolehlivého doručování. Ztrátovost paketů toků VoIP je nicméně téměř nulová, přestože také využívají UDP. Více obrázek 44.



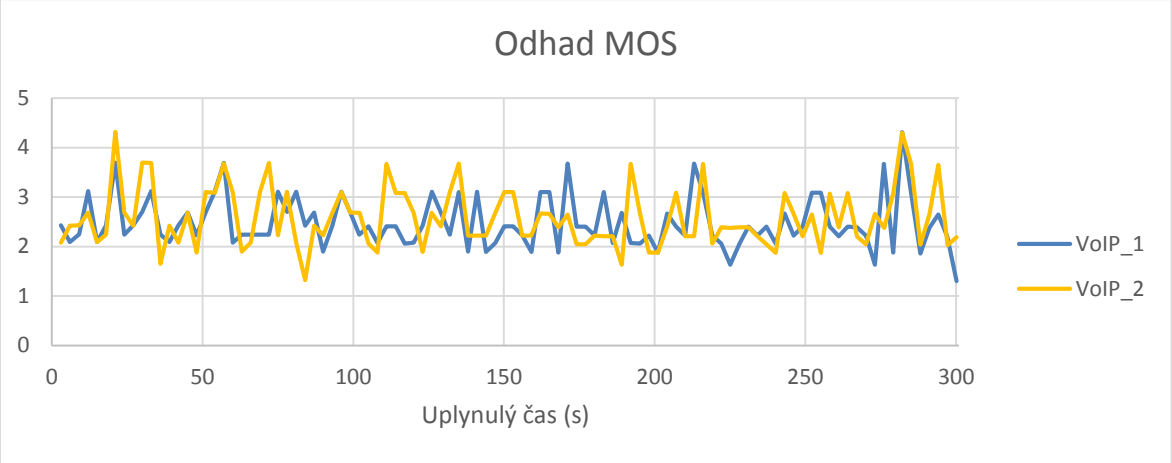
Obrázek 44 – Ztrátovost bez QoS, scénář č. 3

Naměřenené hodnoty hodnoty zpoždění videa jsou již extrémní, toky VoIP by přitom splnily doporučených <150 ms zpoždění (obrázek 45).



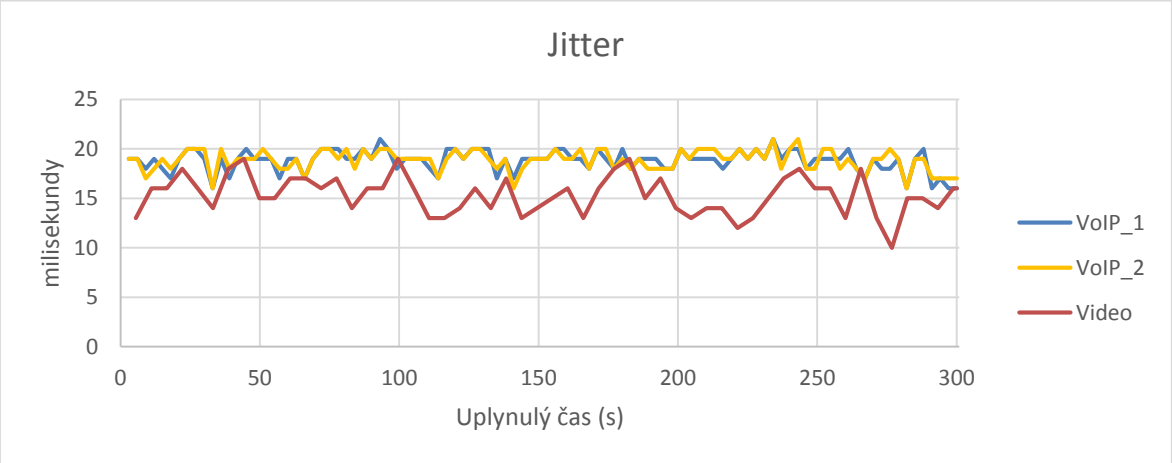
Obrázek 45 – Zpoždění bez QoS, scénář č. 3

I přes stabilně nízké hodnoty zpoždění byl odhad MOS velmi proměnlivý a kvalita komunikace VoIP z hlediska uživatele obecně nízká (obrázek 46).



Obrázek 46 – Odhad MOS bez QoS, scénář č. 3

Zahlcená síť nepřekvapila značně proměnlivými hodnotami jitteru (obrázek 47).



Obrázek 47 – Jitter bez QoS, scénář č. 3

4.3.2 Test sítě s kvalitou služeb

Následuje konfigurace mechanismu AutoQoS. Pakety toků VoIP byly předem značeny pomocí programu IxChariot jako Expedited Forwarding PHB (hodnota DSCP 46). Značení může být prováděno i automaticky zařízením s aktivním AutoQos. Aplikace AutoQoS na konkrétní rozhraní zařízení je velmi snadná:

```
R1> enable
R1# configure terminal
R1(config)# interface Serial0/0/0
R1(config-if)# auto qos voip trust
R1(config-if)# end
```

AutoQoS automaticky vytvoří mapu tříd, s jednou třídou pro samotné tělo komunikace VoIP, třídou pro kontrolní (signalizační) zprávy VoIP a implicitní třídou pro veškerý ostatní síťový provoz. Na směrovači pak vytvořená mapa tříd vypadá následovně:

```
class-map match-any AutoQoS-VoIP-RTP-Trust
 match ip dscp ef
class-map match-any AutoQoS-VoIP-Control-Trust
 match ip dscp cs3
 match ip dscp af31
```

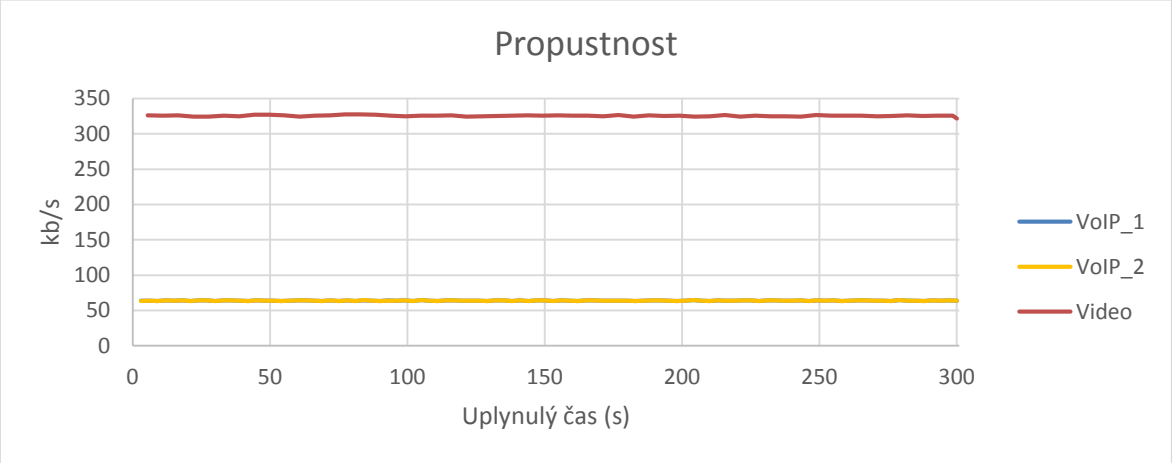
Vytvořené třídy jsou navíc bezpečné v tom smyslu, že pokud stejné značení DSCP polí využívá na zařízení příchozí provoz jiný než VoIP, je hodnota DSCP automaticky přeznačena na DSCP 0 a takový provoz zařazen do třídy implicitní, kde se mu dostane pouze Best Effort zacházení. Označení paketů komunikace VoIP již předem pomocí programu IxChariot posloužilo také jako test, zda je pro AutoQoS takové značení důvěryhodné.

AutoQoS také automaticky vytváří mapu politik:

```
policy-map AutoQoS-Policy-Trust
 class AutoQoS-VoIP-RTP-Trust
  priority percent 70
 class AutoQoS-VoIP-Control-Trust
  bandwidth percent 5
 class class-default
  fair-queue
```

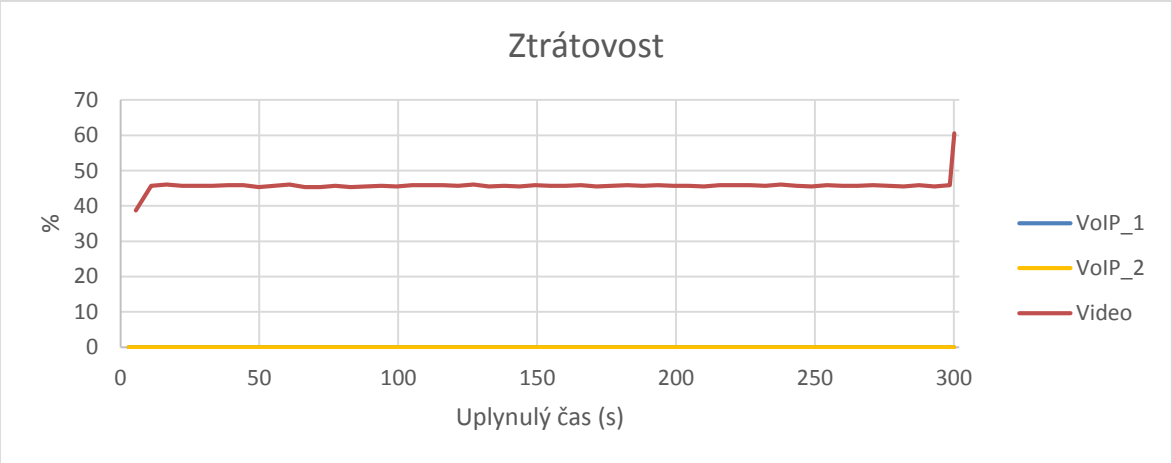
Pro třídu, která ponese tělo komunikace VoIP, je vyhrazeno 70 % šířky pásma v prioritní frontě (LLQ). Signalizaci VoIP je vyhrazeno 5 % z celkové šířky pásma pomocí CBWFQ, tedy bez záruk na zpoždění a jitter. Do implicitní třídy `class-default` spadá veškerý síťový provoz, který nepatří do některé z ostatních 2 tříd. Pro implicitní třídu byla nastavena fronta WFQ, která by měla spravedlivě dělit šířku pásma mezi všechny jednotlivé toky třídy.

Následovalo provedení testu sítě s AutoQoS simulovanou zátěží, nastavení jednotlivých toků zůstává stejné jako při testu bez AutoQoS. Při testu propustnosti se chování sítě prakticky nelišilo od chování sítě bez AutoQoS, viz obrázek 48.



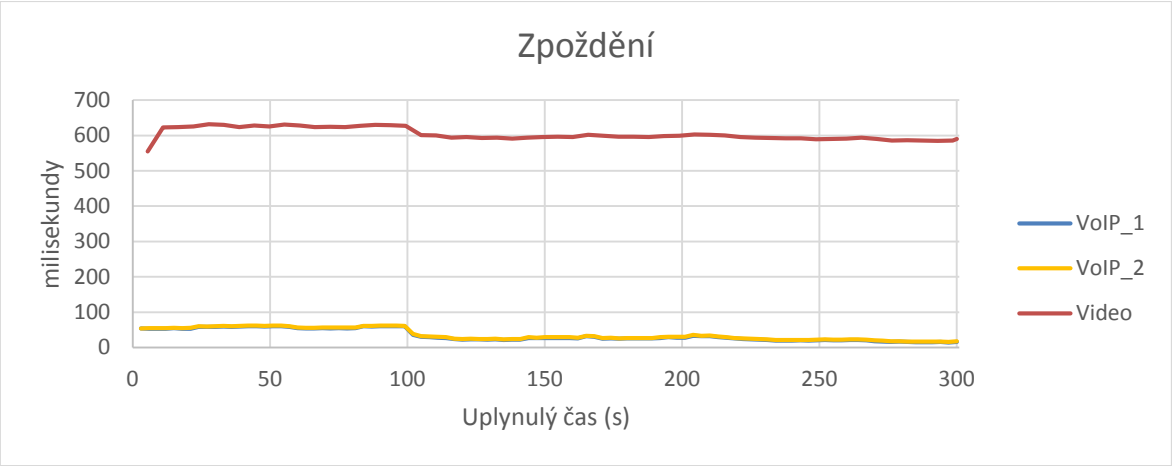
Obrázek 48 – Propustnost s QoS, scénář č. 3

Dalším ze sledovaných parametrů byla ztrátovost. Ztrátovost pro tok videa se jen nepatrně zvýšila oproti případu, kdy v síti nepracovaly žádné nástroje QoS. Ztrátovost toků VoIP zůstala nulová, více obrázek 49.



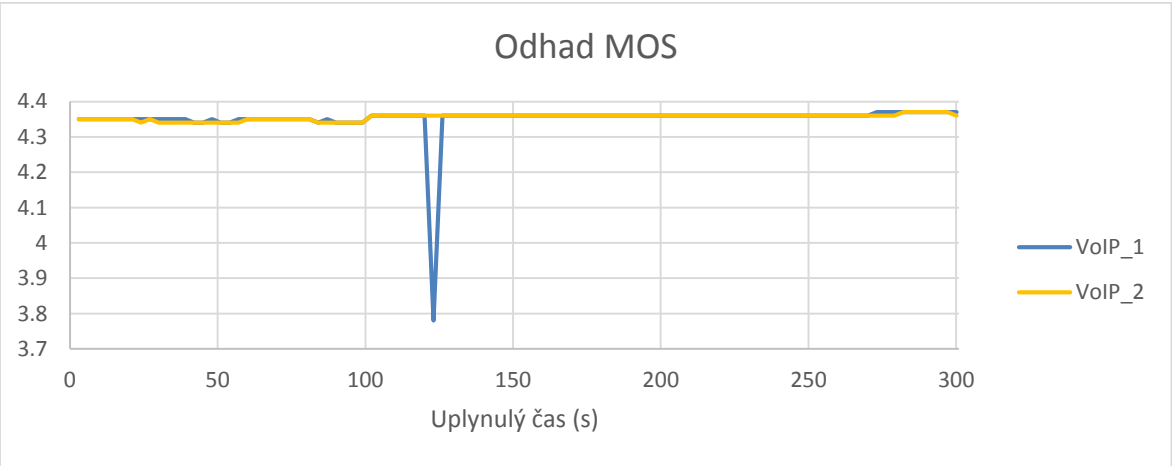
Obrázek 49 – Ztrátovost s QoS, scénář č. 3

U všech toků došlo k obecnému snížení hodnot zpoždění (obrázek 50), což je nicméně pro tok videa na první pohled poněkud překvapující. Fronta, do které byly pakety toku videa umisťovány však byla kratší než ta, do které byly umisťovány bez aktivního AutoQoS. Skutečně přenesené pakety toku videa tak v průměru strávily ve frontě kratší dobu. Více obrázek 50.



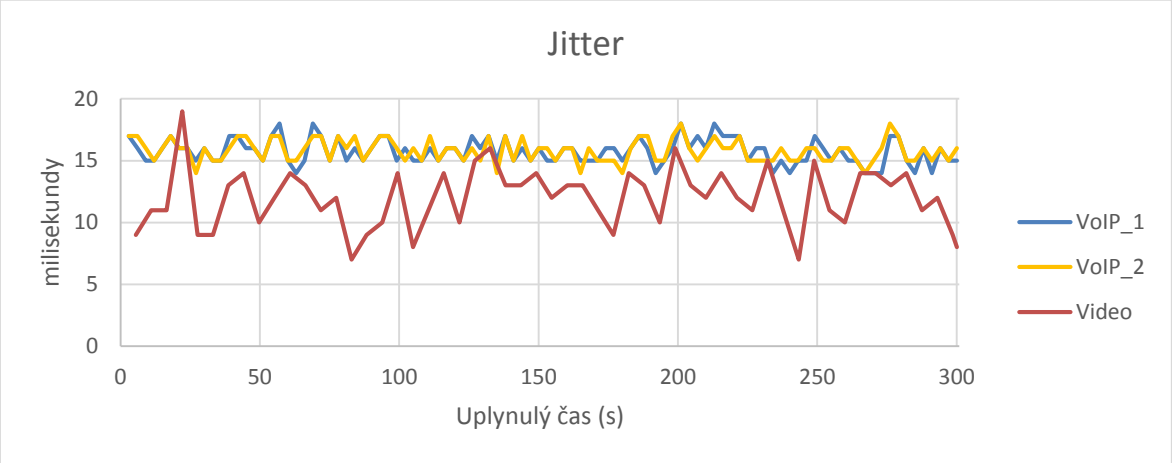
Obrázek 50 – Zpoždění s QoS, scénář č. 3

Odhad MOS (obrázek 51) potvrzuje sílu řešení AutoQoS. Až na jediný výkyv byla po celou dobu kvality komunikace VoIP vynikající, což je v přímém v kontrastu s testem provedeným bez AutoQoS.



Obrázek 51 – Odhad MOS s QoS, scénář č. 3

Posledním ze sledovaných parametrů byl jitter (obrázek 52).



Obrázek 52 – Jitter s QoS, scénář č. 3

4.3.3 Shrnutí výsledků scénáře č. 3

Tabulka v příloze B.3 bakalářské práce přehledně shrnuje dosažené parametry pro jednotlivé toky před a po implementaci AutoQoS. Test potvrdil sílu řešení AutoQoS, kdy jeho triviální aplikací nevyžadující zvláštní znalosti ani časově náročnou konfiguraci bylo dosaženo významného zlepšení kvality komunikace VoIP v testované síti, která by uživatele již plně uspokojila.

5 Závěr

Bakalářská práce se nejdříve teoreticky zabývala mechanismy QoS v sítích IP, v praktické části potom jejich implementací na fyzických zařízeních. To vše s cílem důkladně prostředky QoS prozkoumat a posoudit jejich účinnost a potřebu nasazení. Pro poskytnutí ucelené představy o problematice kvality služeb je nejprve krátce shrnut historický vývoj mechanismů QoS a představeny parametry sítí z pohledu zajištění kvality služeb a také požadavky, které jsou na sítě z tohoto hlediska kladeny. Podrobně jsou diskutovány architektury integrovaných a diferencovaných služeb, rozdíly mezi nimi a techniky, které používají. Pozornost byla zaměřena zejména na architekturu diferencovaných služeb, která sdružuje toky síťového provozu do předem definovaných tříd, neboť architektura integrovaných služeb s sebou nese problém škálovatelnosti ve velkých sítích. Dále jsou diskutovány techniky přepojování paketů s návěstím, které dnes jsou aktuálním prostředím pro nasazení QoS. Do budoucna lze očekávat další růst jejich významu. Na kapitolu o MPLS by se dalo dále navázat prozkoumáním moderních technik MPLS Traffic Engineering, kde se mimo jiné využívá určitého rozšíření rezervačního protokolu RSVP, představeného v práci jako jednu z nepostradatelných částí architektury integrovaných služeb.

V praktické části práce byly nejprve popsány prostředky (jak software, tak fyzické síťové prvky), s nimiž se pracovalo, a poté navrženy tři testovací scénáře. V nich bylo vždy provedeno měření parametrů kvality služeb pro jednotlivé toky při simulované zátěži, kterou generoval program IxChariot. Program WireShark potvrdil korektní značení (a případně přeznačení) hodnot polí DSCP. Nejprve byla síť daného scénáře vždy otestována bez implementace QoS a zhodnoceny dosažené výsledky, které byly porovnány s výsledky dosaženými ve stejné testovací síti s aktivními prostředky QoS.

První scénář se zabýval implementací QoS v doméně MPLS. Jelikož MPLS nepracuje s hlavičkami protokolu IP, které obsahují klíčové informace pro QoS, byl představen princip jejich mapování do pole EXP v návěstí MPLS. Po implementaci QoS se v síti podařilo dosáhnout obecného zlepšení parametrů toků videa a VoIP, kterým byly nástroji QoS vytvořeny vhodné podmínky, zejména umístěním do prioritní fronty. Takové zlepšení však proběhlo na úkor zhoršení parametrů toku FTP. To je důležitý poznatek, neboť daní za preferenci jednoho toku může být zhoršení parametrů toku jiného. Nelze proto QoS nasazovat slepě bez předchozí úvahy.

Druhý scénář představil dvě domény diferencovaných služeb a jejich rozhraní. Scénář představil přeznačení hodnot pole DSCP, které byly považovány za nedůvěryhodné, na hranici jedné z domén. Korektnost postupu potvrdil výstup síťového analyzátoru. Implementací QoS se podařilo dosáhnout zvýšení kvality služby VoIP, na níž se scénář soustředil. Nad tokem videa byl aplikováno omezování. Jeho sílu a správnou funkci prokázaly výsledky propustnosti videa a ztrátovosti jeho paketů, které zahazovač při své činnosti odhazoval.

Původním plánem bylo ve třetím scénáři představit rezervace a funkci protokolu RSVP. Jelikož ale nebyl k dispozici plně funkční RSVP démon, byl předveden nástroj AutoQos firmy Cisco. Ten umožňuje opravdu jednoduché nastavení QoS pro komunikaci VoIP, ke které nejsou třeba žádné rozsáhlé znalosti a je i časově nenáročná. Nastavení navíc zůstává triviální i v komplexní síti, což lze považovat za další velkou výhodu. Dá se očekávat, že automatizace nastavení prostředků QoS se bude nadále rozvíjet a zlepšovat spolu s rostoucí potřebou nasazování QoS.

Potřeba a účinnost nasazení QoS byla výsledky testů jasně prokázána, v zahlcené síti s více druhy síťového provozu se podařilo jeho úpravami pomocí mechanismů QoS vždy zajistit vyšší kvalitu komunikace VoIP (za kritickou nebo kritické aplikace z pohledu kvality služeb lze volit samozřejmě i jinak, dle konkrétních potřeb), což přináší to hlavní, tedy spokojené uživatele.

Literatura

[1] DURAND, B. *Administering Cisco QoS for IP networks*. Editor Michael E. Flannagan. Rockland, Mass.: Syngress Publishing, Inc., c2001, xxiv, 536p. ISBN 1-928994-21-0.

[2] ODOM, W.; CAVANAUGH, M. J. *Cisco QOS exam certification guide: CCVP self-study*. 2nd ed. Indianapolis: Cisco Press, c2005, xxxiv, 730 s. ISBN 15-872-0124-0.

[3] POSTEL, J. *RFC 791 – Internet Protocol Specification.* IETF [online]. 1981 [cit. 2013-02-10]. Dostupný z WWW: <http://www.ietf.org/rfc/rfc791.txt>

[4] NAGEL, J. *RFC 896 – Congestion Control in IP/TCP Internetworks.* IETF [online]. 1984 [cit. 2013-02-10]. Dostupný z WWW: <http://tools.ietf.org/html/rfc896.html>

[5] JACOBSON, V. Congestion avoidance and control. *ACM SIGCOMM Computer Communication Review*. 1995-01-11, roč. 25, č. 1, s. 157-187. ISSN 01464833. DOI: 10.1145/205447.205462. Dostupný z WWW: <http://portal.acm.org/citation.cfm?doid=205447.205462>

[6] STEVENS, W. *RFC 2001 – TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms.* IETF [online]. 1997 [cit. 2013-02-11]. Dostupný z WWW: <http://tools.ietf.org/html/rfc2001>

[7] PARK, K. I. *QOS in packet networks*. New York: Springer Science Business Media, c2005, xii, 243 p. ISBN 03-872-3389-X.

[8] BRADEN, R.; CLARK, D.; SHENKER, S. *RFC 1633 - Integrated Services in the Internet Architecture: an Overview.* IETF [online]. 1994 [cit. 2013-02-11]. Dostupný z WWW: <http://tools.ietf.org/html/rfc1633>

[9] BLAKE, S., et al. *RFC 2475 – An Architecture for Differentiated Services.* IETF [online]. 1998 [cit. 2013-02-11]. Dostupný z WWW: <http://www.ietf.org/rfc/rfc2475.txt>

[10] SANTITORO, R. *Network Service Classes for Simplified QoS Development* [online]. 2003 [cit. 2013-02-12]. Dostupný z WWW: <http://www.itu.int/ITUT/worksem/qos/presentations/qos_1003_s5p5_pres.zip>

[11] JHA, S.; HASSAN, M. *Engineering Internet QoS*. Boston: Artech House, c2002, xx, 325 p. ISBN 15-805-3341-8.

[12] NEOGI, A.; CHIUEH, T.; STIRPE, P. *Performance analysis of an RSVP-capable router.* IEEE Network. 1999, roč. 13, č. 5, s. 56-63. ISSN 08908044. DOI: 10.1109/65.793693. Dostupný z WWW: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=793693>

[13] BRADEN, E., et al. *RFC 2205 – Resource ReSerVation Protocol (RSVP).* IETF [online]. 1997 [cit. 2013-02-13]. Dostupný z WWW: <http://tools.ietf.org/html/rfc2205>

[14] NICHOLS, K., et al. *RFC 2474 – Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers.* IETF [online]. 1998 [cit. 2013-02-11]. Dostupný z WWW: <http://tools.ietf.org/html/rfc2474>

[15] DAVIE, B., et al. *RFC 3246 – An Expedited Forwarding PHB (Per-Hop Behavior).* IETF [online]. 2002 [cit. 2013-02-14]. Dostupný z WWW: <http://tools.ietf.org/html/rfc3246>

[16] BAKER, F., et al. *RFC 5865 – A Differentiated Services Code Point (DSCP) for Capacity-Admitted Traffic.* IETF [online]. 2010 [cit. 2013-02-14]. Dostupný z WWW: <http://tools.ietf.org/html/rfc5865>

[17] HEINANEN, J. *RFC 2597 – Assured Forwarding PHB Group.* IETF [online]. 1999 [cit. 2013-02-14]. Dostupný z WWW: <http://tools.ietf.org/ html/rfc3246>

[18] WANG, Z. *Internet QoS: architectures and mechanisms for quality of service*. San Francisco: Morgan Kaufmann, c2001, xv, 239 s. ISBN 15-586-0608-4.

[19] SZIGETI, T.; HATTINGH, C. *End-to-end QoS network design*. Indianapolis: Cisco Press, 2005, 734 s. ISBN 15-870-5176-1.

[20] ROSEN, E., et al. *RFC 3031 – Multiprotocol Label Switching Architecture.* IETF [online]. 2001 [cit. 2013-02-19]. Dostupný z WWW: <http://www.ietf.org/rfc/rfc3031.txt>

[21] ANDERSSON, L.; ASATI, R. *RFC 5462 – Multiprotocol Label Switching (MPLS) Label Stack Entry: "EXP" Field Renamed to "Traffic Class" Field*. IETF [online]. 2007 [cit. 2013-02-20]. Dostupný z WWW: <http://www.ietf.org/rfc/rfc5036.txt>

[22] ANDERSSON, L., et al. *RFC 5036 – LDP Specification.* IETF [online]. 2007 [cit. 2013-02-20]. Dostupný z WWW: <http://www.ietf.org/rfc/rfc5036.txt>

[23] THE WIRESHARK TEAM. *Wireshark* [software]. 2013 [cit. 2013-02-20]. Dostupný z WWW: <http://www.wireshark.org> Požadavky na systém: 32bitový x86 nebo 64bitový AMD64/x86-64 procesor, operační systém Windows XP Home, XP Pro, XP Tablet PC, XP Media Center, Server 2003, Vista, 2008, 7, or 2008 R2, volné místo na disku 75MB, operační paměť 128 MB, síťová karta

[24] IXIA. *IxChariot* [software]. 2013 [cit. 2013-02-20]. Dostupný z WWW: <http://www.ixchariot.com/products/datasheets/ixchariot.html> Požadavky na systém: Pentium III, operační systém Windows 7 32-bit, Windows 7 64-bit, Windows Vista 32-bit, Windows XP Professional 32-bit, Windows Server 2003 32-

bit, Windows Server 2008 R2 64-bit, volné místo na disku 500 MB, operační paměť 512 MB

[25] Cisco Systems. *Cisco 2800 Integrated Services Routers* [online]. 2013 [cit. 2013-02-20]. Dostupný z WWW: <http://www.cisco.com/en/US/prod/collateral/routers/ps5854/ps5882/product_data_sheet0900aecd8016fa68_ps5854_Products_Data_Sheet.html>

[26] IXIA. *IxChariot Support: KnowledgeBase* [online]. 2013 [cit. 2013-04-08]. Dostupný z WWW: <http://www.ixiacom.com/support/ixchariot/knowledge_base.php>

[27] ITU-T. *G.1010 : End-user multimedia QoS categories* [online]. 2001 [cit. 2013-04-08]. Dostupný z WWW: <http://www.itu.int/rec/T-REC-G.1010-200111-I/en>

[28] ITU-T. *P.800 : Methods for subjective determination of transmission quality* [online]. 1996 [cit. 2013-04-08]. Dostupný z WWW: <http://www.itu.int/rec/T-REC-P.800-199608-I>

Příloha A – Nastavení adresace IP v testovacích scénařích

A.1 Kvalita služby a přepojování paketů s návěstím, AutoQoS

Zařízení	Rozhraní	Adresa IP	Maska podsítě	Výchozí brána
R1	Fa0/0	172.16.1.17	255.255.255.240	N/A
	S0/0/0	192.168.10.1	255.255.255.252	N/A
	Loopback0	1.1.1.1	255.255.255.255	N/A
R2	Fa0/0	172.16.1.33	255.255.255.248	N/A
	S0/0/1	192.168.10.10	255.255.255.252	N/A
	Loopback0	2.2.2.2	255.255.255.255	N/A
R3	S0/0/0	192.168.10.2	255.255.255.252	N/A
	S0/0/1	192.168.10.9	255.255.255.252	N/A
	Loopback0	3.3.3.3	255.255.255.255	N/A
Hostitel 1	NIC	172.16.1.20	255.255.255.240	172.16.1.17
Hostitel 2	NIC	172.16.1.35	255.255.255.248	172.16.1.33

A.2 Kvalita služby a rozhraní domén diferencovaných služeb

Zařízení	Rozhraní	Adresa IP	Maska podsítě	Výchozí brána
R1	Fa0/0	172.16.1.17	255.255.255.240	N/A
	S0/0/0	192.168.10.1	255.255.255.252	N/A
R2	S0/0/0	192.168.10.2	255.255.255.252	N/A
	S0/0/1	192.168.10.5	255.255.255.252	N/A
R3	S0/0/0	192.168.10.9	255.255.255.252	N/A
	S0/0/1	192.168.10.6	255.255.255.252	N/A
R4	Fa0/0	172.16.1.33	255.255.255.248	N/A
	S0/0/0	192.168.10.10	255.255.255.252	N/A
Hostitel 1	NIC	172.16.1.20	255.255.255.240	172.16.1.17
Hostitel 2	NIC	172.16.1.35	255.255.255.248	172.16.1.33

Příloha B – Porovnání výsledků měření před a po implementací kvality služeb

B.1 Kvalita služby a přepojování paketů s návěstím

Scénář	Datový tok	Propustnost			Zpoždění			Jitter			MOS			Ztrátovost
		MIN	MAX	Prům.	MIN	MAX	Prům.	MIN	MAX	Prům.	MIN	MAX	Prům.	Prům.
		kb/s	kb/s	kb/s	ms	ms	ms	ms	ms	ms				%
Bez QoS	FTP	98,1	388,9	124,4	-	-	-	-	-	-	-	-	-	-
	Video	391,9	400,8	398,8	75	681	575	1	2	1	-	-	-	0.061
	VoIP	62,8	64,2	63,7	65	672	566	7	9	7,9	1,0	4,34	2,43	0.147
S QoS	FTP	68,8	148,3	84	-	-	-	-	-	-	-	-	-	-
	Video	399	425,1	400	38	66	63	1	1	1	-	-	-	0.03
	VoIP	63,9	70,5	64	31	58	54	8	8	8	4.35	4.36	4.35	0

B.2 Kvalita služby a rozhraní domén diferencovaných služeb

Scénář	Datový tok	Propustnost			Zpoždění			Jitter			MOS			Ztráto vost
		MIN	MAX	Prům.	MIN	MAX	Prům.	MIN	MAX	Prům.	MIN	MAX	Prům.	Prům.
		kb/s	kb/s	kb/s	ms	ms	ms	ms	ms	ms	-	-	-	%
Bez QoS	FTP	191,1	426,1	215,2	-	-	-	-	-	-	-	-	-	-
	Video	394,8	401,1	399,6	67	321	240	7	8	7,69	-	-	-	0
	VoIP	62,9	64,1	63,9	57	310	230	1	1	1	3,35	4,34	3,76	0
S QoS	FTP	429,8	451,6	440,1	-	-	-	-	-	-	-	-	-	-
	Video	241	252,3	242	46	65	57	0	1	0,49	-	-	-	39,4
	VoIP	63,5	64,4	63,9	28	49	40	15	17	16,2	4,35	4,36	4,35	0

B.3 AutoQoS

Scénář	Datový tok	Propustnost			Zpoždění			Jitter			MOS			Ztráto vost
		MIN	MAX	Prům.	MIN	MAX	Prům.	MIN	MAX	Prům.	MIN	MAX	Prům.	Prům.
		kb/s	kb/s	kb/s	ms	ms	ms	ms	ms	ms	-	-	-	%
Bez QoS	Video	318,5	326,5	325,2	1276	2113	2066	10	19	15,3	-	-	-	45,7
	VoIP_1	63,09	64,90	63,97	92	112	102	16	21	18,8	1,30	4,31	2,48	0,01
	VoIP_2	63,24	63,97	64,62	95	116	105	16	21	18,8	1,32	4,32	2,58	0,01
S QoS	Video	321,5	327,5	325,5	555	632	604	7	19	12	-	-	-	45,8
	VoIP_1	63,3	64,8	63,9	14	60	35	14	18	15,7	3,78	4,37	4,35	0
	VoIP_2	63,2	64,8	63,9	16	62	36	14	18	15.8	4,34	4,37	4,36	0

Příloha C – Kontrola značení pole DSCP pomocí síťového analyzátoru

C.1 Přeznačení PHB AF 41 na PHB AF 31