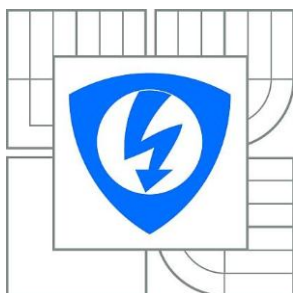


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

PODPORA KVALITATIVNÍCH POŽADAVKŮ SLUŽEB V OPERAČNÍCH SYSTÉMECH UNIXOVÉHO TYPU PRO PROVOZ V BEZDRÁTOVÝCH SÍTÍCH WIFI

QUALITY OF SERVICE SUPPORT IN UNIX-LIKE OPERATING SYSTEMS FOR COMMUNICATION
IN WIFI NETWORKS

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

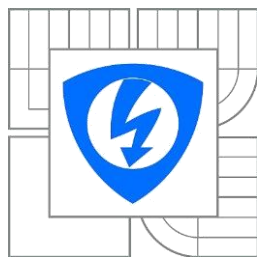
AUTOR PRÁCE
AUTHOR

Bc. JOSEF MIZERA

VEDOUCÍ PRÁCE
SUPERVISOR

doc. Ing. VÍT NOVOTNÝ, Ph.D.

BRNO 2014



**VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ**

**Fakulta elektrotechniky
a komunikačních technologií**

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor
Telekomunikační a informační technika

Student: Bc. Josef Mizera

ID: 119538

Ročník: 2

Akademický rok: 2013/2014

NÁZEV TÉMATU:

Podpora kvalitativních požadavků služeb v operačních systémech unixového typu pro provoz v bezdrátových sítích WiFi

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s technologiemi řešení problémů podpory kvalitativních požadavků služeb v datových sítích. Zaměřte se na řešení podpory v koncových zařízeních, konkrétně datových terminálech s operačním systémem unixového typu. Detailně prostudujte techniku vícenásobného přístupu v bezdrátových sítích dle skupiny standardů IEEE 802.11 včetně techniky podpory QoS. Navrhněte měřicí pracoviště a proveďte řadu měření s různými profily služeb a s různým nastavením komunikačních parametrů. Výsledky měření vyhodnoťte a navrhněte sadu doporučení pro nastavení parametrů v terminálech dle předpokládaného profilu služeb. Navrhněte laboratorní úlohu pro předmět Architektura sítí a vypracujte k ní návod.

DOPORUČENÁ LITERATURA:

- [1] G. SNYDER, T. R. HEIN, E. NEMETH: Linux - Kompletní příručka administrátora. Computer press, ISBN 9788025124109, ČR, 2008
- [2] MARCHESE M. QoS over Heterogeneous Networks. John Wiley & Sons, ISBN 978-0-470-01752-4, 2007

Termín zadání: 10.2.2014

Termín odevzdání: 28.5.2014

Vedoucí práce: doc. Ing. Vít Novotný, Ph.D.

Konzultanti diplomové práce:

doc. Ing. Jiří Mišurec, CSc.

UPOZORNĚNÍ:

Předseda oborové rady

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

Anotace

Diplomová práce je zaměřena na podporu kvalitativních požadavků služeb v bezdrátových sítích, především v operačních systémech Unixového typu. Téma je spojeno nejen se samotným OS, ale také se standardem bezdrátových sítí podporujícím QoS, který je nazván IEEE 802.11e. Podpora QoS je zapotřebí zejména u časově náročných datových přenosů v reálném čase.

Teoretická část práce se zabývá teoretickým rozbořem problematiky podpory QoS. Jsou zde popsány parametry, které se u řešení podpory kvalitativních požadavků služeb vyskytují. Tato část se také zabývá dělením služeb používaných pro přenos dat v počítačových sítích. Je zde také popsána podpora QoS v bezdrátových sítích podle standardu 802.11e, jeho zavedení a metody přístupu k médiu s možností klasifikace provozu a bez možnosti klasifikace provozu. Na tuto část navazuje popis podpory QoS v operačních systémech unixového typu. V kapitole je popsáno, jakým způsobem je podpora QoS v těchto operačních systémech řešena, dále jsou zde charakterizovány jednotlivé nástroje, které se pro řízení datového toku v operačních systémech linux používají. V poslední řadě se teoretická část zabývá jednotlivými typy front a metodami používanými v OS linux.

V praktické části práce jsou navrženy různé topologie a scénáře, pro ověření funkčnosti podpory QoS v bezdrátových sítích s využitím unixového systému. V těchto kapitolách jsou zobrazeny výsledky různých testování při přenosu zvolených datových toků, které jsou citlivé na přenos v čase. Je zde také ověřena součinnost podpory QoS mezi zařízeními fungujícími na síťové a linkové vrstvě.

Výstupem práce je navržení laboratorní úlohy pro předmět Architektura sítí. Tato úloha je zaměřena na seznámení se s funkčností podpory QoS v bezdrátových sítích a v OS unixového typu. V kapitole jsou popsány samotná zařízení a programy, které jsou pro změření úlohy potřeba. Poslední část kapitoly popisuje postup pro přípravu měřicího pracoviště. K laboratorní úloze je v příloze vypracován návod k vypracování.

Klíčová slova

QoS, VoIP, HTB, HFSC, WiFi, UNIX, služba, propustnost, ztrátovost, zpoždění, kolísavost, šířka pásma

Abstract

Diploma thesis is focused on the supporting of Quality of Services in wireless networks, especially in the Linux operation systems. The topic is connected not only with OS, but also with the wireless standard, which supports QoS in wireless networks called IEEE 802.11e. QoS is needed especially for time-consuming data transfers in real time.

The theoretical part deals with the theoretical analysis of the issue of the QoS support. There are described parameters, which occurred in quality of services support. This section also deals with the division of services that are used to transmit data across computer networks. It also describes the QoS support in wireless networks according 802.11e, its implementation and methods of accessing a medium with a without possibility of traffic. This part is followed by a description of QoS support in UNIX operating systems. The chapter describes how is the QoS support designed in these operating systems. There are also characterized concrete tools which are used for control the data flow in the operating systems using Linux. At the end the theoretical part deals with different types of queues and methods used in linux OS.

In the practical part of the thesis, there are various designed topologies and scenarios to verify the functionality of QoS support in wireless networks using a Unix system. These chapters show the results of different tests at selected transmission data streams that are sensitive to transmission time. There is also verified cooperation of QoS support between devices operating on the network and data link layers.

The output of this work is to design a laboratory exercise for the subject Network Architecture. This exercise is focused on familiarization with the QoS support functionality in wireless networks and in Unix-like operating systems. This chapter also describes the devices and programs that are needed to measure this task. The last part of the chapter describes the procedure for the preparation of the measuring station. For this laboratory task, there is an inserted manual in the annex.

Key words

QoS, VoIP, HTB, HFSC, WiFi, UNIX, service, throughput, packet-loss, delay, jitter, bandwidth

Bibliografická citace:

MIZERA, J. *Podpora kvalitativních požadavků služeb v operačních systémech unixového typu pro provoz v bezdrátových sítích WiFi*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2014. 83 s. Vedoucí diplomové práce doc. Ing. Vít Novotný, Ph.D..

Prohlášení

Prohlašuji, že svoji diplomovou práci na téma „Podpora kvalitativních požadavků služeb v operačních systémech unixového typu pro provoz v bezdrátových sítích“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb..

V Brně dne

.....

podpis autora

Poděkování

Děkuji vedoucímu práce doc. Vítu Novotnému za užitečnou metodickou pomoc a cenné rady při zpracování diplomové práce.

V Brně dne

.....

podpis autora

PODĚKOVÁNÍ

Výzkum popsáný v této diplomové práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

Brno

.....
(podpis autora)

Obsah

1. Úvod.....	11
2. QoS	12
2.1 Parametry QoS	12
2.2 Služby používané pro přenos dat v sítích	13
3. Technologie 802.11 a WiFi.....	14
3.1 Zavedení 802.11e	14
3.2 DCF & PCF	15
3.3 WMM.....	16
3.3 EDCF & HCF	18
3.3.1 Enhanced Distributed Coordination Function (EDCF)	19
3.3.2 Hybrid coordination function (HCF)	20
4. Podpora QoS v operačních systémech Unixového typu	21
4.1 Hierarchické sdílení média	21
4.2 Řízení datového toku	22
4.2.1 Shaping (tvarování).....	22
4.2.2 Scheduling (plánování)	23
4.2.3 Classifying (klasifikace)	23
4.2.4 Policing	24
4.2.5 Dropping.....	24
4.2.6 Marking.....	24
4.2.7 QDisc.....	25
4.2.8 Class.....	25
4.2.9 Filter.....	25
4.2.10 Classifier	26
4.2.11 Policar	26
4.2.12 Drop.....	26

4.2.13	Handle	27
5.	Fronty.....	27
5.1	Classless	28
5.1.1	FIFO.....	28
5.1.2	pfifo_fast	28
5.1.3	SFQ.....	29
5.1.4	ESFQ.....	30
5.1.5	GRED	30
5.1.6	TBF	30
5.2	Classfull.....	31
5.2.1	PRIO	31
5.2.2	CBQ.....	32
5.2.3	HTB	35
5.2.4	HFSC.....	38
6.	QoS v OS Unixového typu.....	45
6.1	Topologie se dvěma stanicemi, z nichž jedna slouží jako AP	45
6.2	Topologie se třemi stanicemi, z nichž jedna slouží jako AP	53
6.3	Topologie vytvoření sítě pomocí Access Pointu.....	56
6.4	Topologie vytvoření sítě se dvěma Access Pointy.....	61
7.	Laboratorní úloha	67
	Závěr	70
	Seznam použité literatury:.....	71
	Seznam obrázků:	72
	Seznam tabulek:.....	74
	Seznam rovnic:.....	74
	Seznam příloh:	74
	Seznam zkratk:	75

1. Úvod

Diplomová práce je z velké části zaměřena na bezdrátové sítě. Bezdrátová síť WLAN (Wireless Local Area Network) je jeden z typů počítačových sítí. Tyto sítě využívají ke spojení jednotlivých účastníků bezdrátovou komunikaci, ve většině případů pomocí elektromagnetických vln. Bezdrátové sítě WLAN, jsou často také nazývané WiFi (Wireless Fidelity), což ale jsou standardy vydané organizací IEEE, které jsou využívány pro bezdrátovou komunikaci v sítích WLAN. Tyto sítě jsou v dnešní době jedním z nejpoužívanějších způsobů pro připojení k internetu. Tato práce bude zaměřena na nejpoužívanější standardy ve WLAN sítích, dále na pravidla komunikace při přenosu multimediálních dat a na upřednostnění tohoto přenosu před ostatními druhy přenosu. Celý projekt bude zaměřen na podporu kvalitativních požadavků služeb QoS (Quality of services) v operačních systémech Unixového typu.

Při přenosu dat po Internetu jsou nejvíce kladené nároky na přenos multimediálních dat. To je způsobeno tím, že se počet uživatelů zařízení, která dokáží tyto přenosy uskutečnit, velmi rychle zvyšuje. V nejčastějším případě se jedná o zařízení, se kterými je možno provádět např. videohovory, videokonference, streamování videí, online přenosy, VoIP hovory,

U zařízení využívajících připojení s technologií Ethernet, není nastavení požadovaných priorit příslušným médiím tak složité, jak je tomu u zařízení připojených pomocí bezdrátových sítí. Bezdrátové sítě nedisponují tak širokou kapacitou linky jak je tomu u linek s drátovým připojením. Z tohoto důvodu se zavádějí mechanismy, které dokáží priority médiím v datovém přenosu přiřadit.

Metoda, která je běžně používaná pro přenos paketů v internetu se nazývá best effort. Best effort je metoda nejvyšší snahy, tzn., nevyužívá žádnou službu podpory QoS. Pokouší se o správné doručení paketů v co nejkratším čase. Podpora kvalitativních požadavků služeb QoS se skládá ze čtyř hlavních parametrů kolísání zpoždění (jitter), šířka pásma (bandwidth), zpoždění (delay) a ztrátovost paketů (packet loss). První část práce bude popisovat stručný popis těchto parametrů, včetně popisu oblastí přenosu, kde je nejvíce třeba tyto parametry aplikovat.

Důležitým prvkem v oblasti vývoje pro aplikování podpory QoS do datových sítí byl vývoj standardů 802.11. Prvním standardem IEEE 802.11, který plně podporuje nastavení požadavků pro přenos multimediálních dat je standard IEEE 802.11e, kterému bude také věnována část této práce.

K implementování QoS pro datový přenos je třeba zvolit vhodné metody, které budou plně funkční v operačních systémech založených na jádře Linuxu. V práci budou popsány vhodné metody pro implementaci QoS v OS Unixového typu a v následující části budou navrženy scénáře pro otestování funkčnosti nejvhodnějších metod podpory QoS při přenosech multimediálních dat.

2. QoS

Quality of Service je termín, který se používá pro technologie řídicí datový tok v sítích, které využívají technologii přepojování paketů. QoS musí umět rozlišit druh přenosu, aby pomocí této informace mohl nastavit požadovanou prioritu danému typu datového přenosu. Priorita pro daný typ přenosu udává to, že se pakety důležitějšího typu přenosu dostanou do stanice příjemce ve správnou dobu a nepoškozeny [1].

Nejvíce žádané nastavení QoS je u přenosů v reálném čase, tzv. Real-time přenosů. U přenosů Real-time je třeba mít nastaveny priority tak, aby při přenosu dat nedocházelo ke ztrátě a zpoždění dat.

Real-time služby nepožadují pouze bezchybný přenos dat, ale hlavně přenos dat v požadované časové relaci. Nejdelší hranice pro přenos dat se nazývá deadline. U přenosů v reálném čase je možno ztratit některé pakety, ale je nežádoucí, aby pakety došly v jiném pořadí a se zpožděním [2].

2.1 Parametry QoS

Nastavení parametrů QoS musí být provedeno tak, aby oblasti přenosu, které nějak omezují či mění jeho kvalitu a rychlost, byly uvedeny do takového stavu, že v případě obsazené linky budou tyto oblasti pro daný typ přenosu mít konstantní hodnoty. Jedná se o tyto parametry:

- Delay (zpoždění) – při datovém přenosu vždy dochází k určitému zpoždění. Nasazení podpory QoS zajišťuje to, že toto zpoždění bude nižší než maximální akceptovatelná hodnota daného typu služby. Zpoždění je ovlivněno několika podněty. Prvním podnětem je propagace, která je udána rychlostí šíření signálu přenosovým médiem. Jako další je serializace, která říká jakou rychlostí je linka schopna pojmout vkládání dat. Dále to může být zpracování, to jsou operace zpracování dat např. ve směrovačích, zdržení ve frontách, buffer, a další.
- Jitter (kolísání) – při přenosu dat mají jednotlivé pakety rozdílné zpoždění, které je převážně způsobeno frontami v přenosové cestě, případně i rozdílnými trasami paketů.
- Packet loss (ztrátovost) – při přenosu dat dochází občas k tomu, že je některý z přenášených paketů ztracen nebo zahozen kvůli poškození. U sítí typu TCP musí být tento paket zaslán znovu, což ale způsobí změnu pořadí paketů. Tento fakt nevádí při přenosu určitého typu dat, protože data jsou v cílové stanici opět přeuspořádána do správného pořadí. U přenosů dat v reálném čase je tato možnost nepřipustná, proto je zde používán nespolehlivý protokol UDP, ve kterém není implementován mechanismus pro opětovné zaslání paketu. U přenosů v reálném čase např. hlasu a videa, je ve většině případů tolerována určitá ztráta paketů.
- Throughput (propustnost) – množství informace, které může projít spojem v reálné situaci.

- Out of order delivery (doručení mimo pořadí) – datové pakety mohou použít jinou cestu nebo mohou být při přenosu ztraceny.
- Bandwidth (šířka pásma) – množství dat, které může být fyzicky přeneseno přes vybrané médium. Tato hodnota je teoreticky vypočítána a v reálné situaci nemůže být dobře dosažena, protože aktuálně není taková propustnost dostupná [2].

U přenosů v reálném čase existují doporučené hodnoty těchto parametrů pro plynulý přenos hlasu nebo videa. Příklad doporučených hodnot parametrů QoS pro internetovou telefonii VoIP:

- Latence (doba mezi vysláním a doručení paketu) < 150 ms.
- Jitter (rozdíl intervalů přijímaných paketů) < 30 ms.
- Packet loss (podíl přijatých a poslaných paketů za čas) > 99%.
- Bandwidth (souvisí s propustností) 12 – 106 kb/s v závislosti na vzorkování, kodeku a režii na 2. vrstvě [1].

2.2 Služby používané pro přenos dat v sítích

Tyto služby jsou děleny dle toho, jaký druh podpory QoS používají, nebo zda vůbec nějaký používají.

- Best effort service – služba nepoužívající podporu QoS vůbec, tato služba je využívána u klasického TCP/IP provozu, doba doručení paketů není specifikována a je proměnná v závislosti na zatížení sítě.
- Garantované služby – služby zaručující doručení všech odeslaných dat. Tyto služby jsou vždy dostupné.
- Služby s řízením zátěže – garantují průměrné zpoždění a doručení téměř všech přenášených dat, ne ale s takovou garancí jak je tomu u služeb garantovaných [2].



OBRÁZEK 2.1 - DĚLENÍ SLUŽEB QOS Z HLEDISKA KOMPLEXNOSTI

- Best Effort – nemění pořadí paketů
- Diffserv – rozlišuje druhy datových toků a přiřazuje jim různé priority
- IntServ – rezervuje pevné šířky pásma pro různé aplikace [3].

3. Technologie 802.11 a WiFi

Bezdrátové standardy 802.11 jsou jedním z nejvíce vyvíjených odvětví síťových technologií. První standard 802.11 byl představen v USA roku 1997 organizací IEEE (The Institute of Electrical and Electronics Engineers). Standardy 802.11 specifikují bezdrátové sítě, které byly nejprve uplatňovány v místech, kde bylo složité přivést síťové připojení jinou cestou a to buď s technických, nebo ekonomických důvodů. Později se tyto sítě začaly využívat daleko více, protože poskytovaly uživateli mobilitu, která je hodně oceněná. V dnešní době jsou tyto sítě velice rozšířené a jsou dostupné téměř všude v civilizovaném světě. S tímto také přichází zvyšování nároků od uživatele na kvality těchto sítí. S tím, že se začali ve velkém množství využívat multimediální přenosy typu VoIP, videokonverzace, atd., přišel také požadavek uskutečnit tyto přenosy v reálném čase v bezdrátových sítích. To je možné řešit pomocí implementace QoS. Organizace IEEE vyvinula standard bezdrátových sítí, který podporu kvalitativních požadavků služeb uskutečňuje, standard je nazýván 802.11e. Předchůdcem standardu 802.11e byla služba WMM (Wireless MultiMedia). 802.11e přinesl vylepšení přístupu k MAC vrstvě a také definuje nové mechanismy pro podporu QoS ve WLAN. Jsou to služby EDCF (Enhanced Distributed Coordination Function - zvýšená distribuovaná koordinační funkce) a HCF (Hybrid Coordination Function - hybridní koordinační funkce) [4].

3.1 Zavedení 802.11e

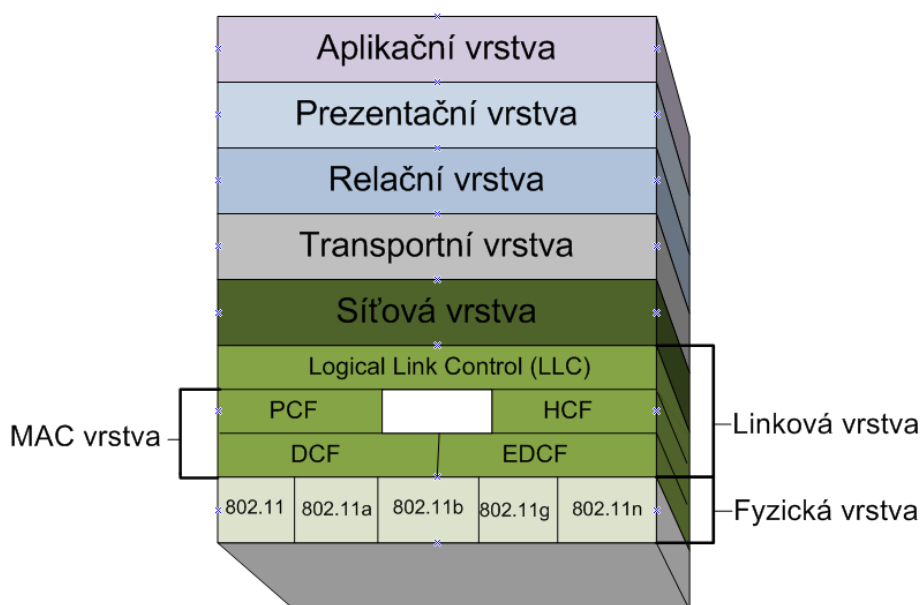
Sítě WLAN donutily vývojáře komunikačních standardů především IEEE k mnoha inovacím. Dva nejrozšířenější standardy 802.11b/802.11g, využívají převážně protokol 802 LLC (Logical Link Control), který má na starosti fyzický přenos dat na přenosové médium, také poskytuje nezávislou fyzickou vrstvu a podvrstvu přístupu k médiu MAC, která umožňuje bezdrátový přenos typu Best Effort snažící se doručovat datové pakety k příjemci v co nejkratším čase [4].

Při zavádění sítí WLAN se v určitých místech očekává úplné nahrazení kabelových sítí. Nahrazováním kabelových sítí došlo také k tomu, že se zvýšily požadavky na bezdrátové sítě. Je požadováno, aby tyto sítě poskytovaly stejné služby jako sítě LAN. Při přetížení přenosové linky se může kvalita některých přenosů dostat do nepřijatelných hodnot, což může být u sítí WLAN způsobeno omezenou šířkou pásma. Pro bezdrátové protokoly WLAN, bylo třeba vyvinout metodu pro efektivní využití přenosového kanálu, vyhnout se kolizím a spravedlivě rozdělit limitovanou šířku pásma mezi individuální typy přenosů v IP sítích, které jsou rozděleny podle výšky priority tohoto přenosu.

DCF (Distributed Coordination Function - distribuovaná koordinační funkce) a PCF (Point Coordination Function - bodová koordinační funkce) v původní MAC vrstvě protokolu IEEE 802.11

neposkytovaly služby pro rozlišovací mechanismy, které zaručují dolní mez propustnosti a horní mez ztrátovosti paketů [4].

Skupina IEEE vyvinula, na základě požadavků pro přenos audio a video dat, MAC protokol, který rozlišuje druh přenášených paketů a dokáže přidělovat priority paketům. 802.11e zavádí dva nové režimy pro vrstvu MAC, které poskytují podporu QoS pro žádající stanice. Režimy jsou známy pod názvy EDCF a HCF.



OBRÁZEK 3.1 - VRSTVOVÝ MODEL OSI S MODIFIKACEMI 802.11 A 802.11E [5]

3.2 DCF & PCF

Metoda DCF využívá mnohonásobného přístupu k médiu s předcházením kolize CSMA / CA (Carrier sense Multiple Access with Collision Avoidance). Jsou-li ve vysílací stanici připravena data pro odeslání a vysílací stanice zjistí, že je kanál volný, přenosová stanice čeká po dobu DCF Inter – Frame Space (DIFS), což je minimální požadovaný klidový stav. V případě že se na médium nepokusí připojit jiná stanice, může začít přenos. Pokud na počátku bude kanál obsazený nebo v případě, že se během čekací doby DIFS pokusí připojit jiná stanice přichází na řadu metoda předcházení kolizím. U této metody se po uplynutí času DIFS čeká po dobu náhodně vybraného počtu časových slotů (backoff counter) z (0, 1, ..., CW-1). Hodnota CW (contention window – okno sváru) je z počátku nastavena na hodnotu CWmin. V případě, že nastane kolize, se hodnota CW zdvojnásobuje. Tento cyklus probíhá do doby, dokud hodnota CW nenabude hodnoty CWmax. Pokud je nalezen kanál, volný po dobu DIFS, backoff counter začne snižovat hodnotu jednomu z volných časových slotů. Nastane-li situace, že je kanál obsazen, backoff counter je pozastaven do opětovného uvolnění kanálu po dobu, kterou byl volný v minulém DIFS. Jakmile je hodnota v backoff counteru rovna nule, dochází k zahájení přenosu dat.

DCF přenos poskytuje stanicím rovnoměrnou pravděpodobnost přístupu ke kanálu se sdíleným médiem, dle shodné rámcové frekvence, ale s různou propustností. Tato spojení nejsou vhodná pro přenos dat v reálném čase, protože neposkytují podporu pro přiřazení priorit danému typu dat.

Metoda PCF je známá jako první metoda, která se snaží podporovat časově náročné datové přenosy. Bodový koordinátor (PC – point coordinator) pravidelně vysílá rámec beacon identifikátorům vysílací sítě a řídicím parametrům, které jsou specifické pro bezdrátové sítě. PCF rozdělí čas na období bez spojení (CFP – contention free period) a na období se spojením (CP – contention period). Vysílat může pouze ten počítač, který vyzve stanici k vysílání a zároveň tato výzva musí být v době CFP, která udává volnost linky. CFP končí po uplynutí doby, která je uvedena v rámci beacon, nebo po přijetí koncového rámce CF- End Frame. Metoda PCF sice nabízí určitou prioritu přetíženým stanicím, ale nenabízí možnost rozlišení zdroje a typu provozu. Nelze tedy definovat, které stanice mají prioritu dlouhého časového přenosu a které mohou využít pouze metodu Best Effort. Dalším problémem u PCF je, že se počítačové stanice musí potýkat s ostatními stanicemi o získání kontroly nad bezdrátovým médiem. Proto se může lišit čas začátku a délka CFP. Tyto důvody vedly skupinu IEEE k založení standardu 802.11e [4].

3.3 WMM

Wi-Fi multimédia s původním názvem Wireless Multimedia Extension (WME), spadá pod standard bezdrátových sítí WLAN 802.11e. Tato služba poskytuje specifikaci pro zlepšení kvality služeb implementací podpory QoS do sítí WLAN. Zlepšení kvality přenosu služeb spočívá v rozdělení datových paketů do čtyř kategorií. Každá z těchto kategorií má přidělenou určitou prioritu přenosu viz tabulka 3.1 [6].

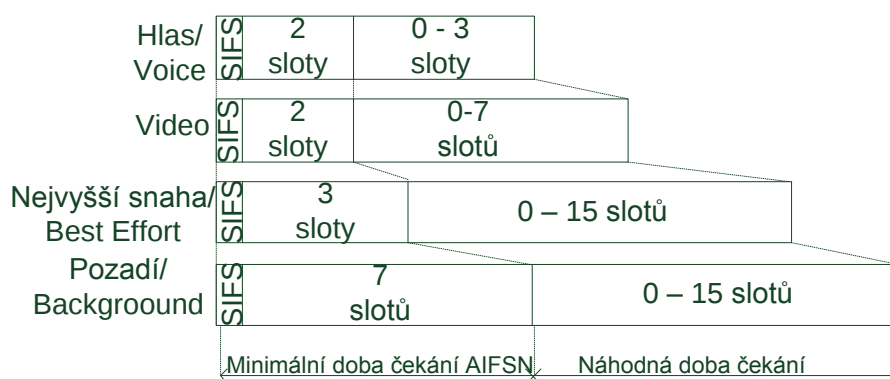
TABULKA 3.1- KATEGORIE PŘÍSTUPU WMM

třída přístupu	popis	výška priority dle 802.1D
Priorita Hlasu	Nejvyšší priorita: umožní více souběžných hovorů VoIP s nízkým zpožděním a vysokou kvalitou zvuku	7,6
Priorita Videu	Upřednostní přenos videa před ostatními datovými toky. Jeden přístupový kanál v síti 802.11 g nebo 802.11a může podporovat 3-4 SDTV streamy nebo jeden HDTV stream.	5,4
Priorita Best Effort	Přenos z ostatních zařízení nebo aplikací, které nevyžadují QoS. Přenosy s nižší citlivostí na zpoždění. Např. internetový prohlížeč.	0,3
Ostatní priority na pozadí (background)	Nejnižší priorita: tiskové úlohy, přenos dat, přenosy, které nemají přísné požadavky na zpoždění a propustnost	1,2

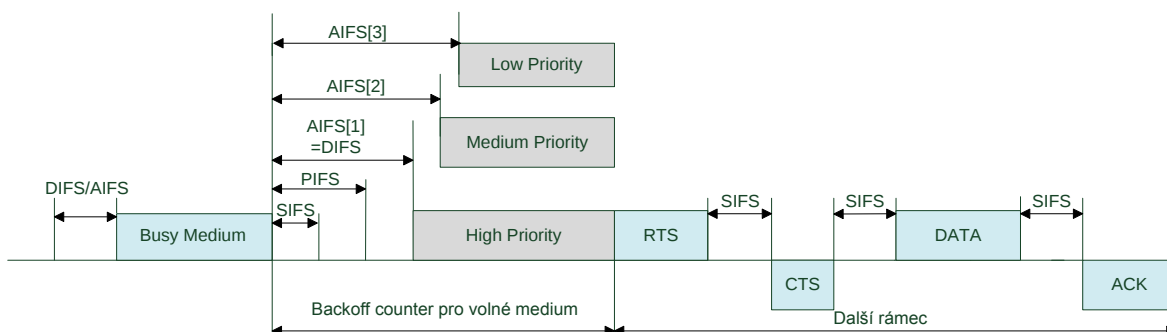
WMM pracuje podobně jako metoda DCF. Nevyužívá pouze čekací dobu DIFS, ale pracuje s čekací dobou AIFS (Arbitration Inter-Frame Space), která se při zvyšování priority zkracuje. Díky tomu budou mít data, kterým je přiřazena nejvyšší priorita nejrychlejší přístup k médiu. AIFS se skládá z hodnoty AIFSN (Arbitration Inter-Frame), složené z násobků slot time a hodnoty SIFS (Short Inter-Frame Space), dále je do AIFSN přičten náhodný interval CW, který je proměnný. Velikost hodnoty AIFSN se odvíjí od třídy přístupu. Doby čekání pro přístup k přenosovému médiu pro jednotlivé priority jsou na obrázku 3.2 a v tabulce 3.1 [6].

Přidělení priority závisí na časovacích parametrech, které se liší u každé přístupové kategorie (AC – access categories):

- Minimální mezirámcový prostor nebo libovolné mezirámcové prostorové číslo AIFSN
- Hodnotě CW nebo náhodné čekací doby backoff
- CWmin, CWmax, AIFS a limit TXOP (transmission opportunity) pro každou AC distribuovanou v rámcích beacon



OBRÁZEK 3.2 - WMM DOBA PŘÍSTUPU K MÉDIU U JEDNOTLIVÝCH TŘÍD PŘENOSU



OBRÁZEK 3.3 - DCF OKAMŽITÝ PŘÍSTUP PRO PRVNÍ RÁMEC, KDYŽ JE MÉDIUM VOLNÉ (DIFS/AIFS[1])

TABULKA 3.2 - ČASY PŘÍSTUPU K MÉDIU U RŮZNÝCH STANDARDŮ [7]

AC	AIFSN	802.11b AIFS[AC]	802.11g AIFS[AC]	802.11a AIFS[AC]	802.11n 2,4GHz AIFS[AC]	802.11n 5GHz AIFS[AC]
SIFS Time	-	10μs	10μs	16μs	10μs	16μs
Slot Time	-	20μs	dlouhý = 20μs krátký = 9μs	9μs	dlouhý = 20μs krátký = 9μs	9μs
Priorita Hlasu	2	50μs	dlouhý = 50μs krátký = 28μs	34μs	dlouhý = 50μs krátký = 28μs	34μs
Priorita Videa	2	50μs	dlouhý = 50μs krátký = 28μs	34μs	dlouhý = 50μs krátký = 28μs	34μs
Priorita Best Effort	3	70μs	dlouhý = 70μs krátký = 37μs	43μs	dlouhý = 70μs krátký = 37μs	43μs
Ostatní priority	7	150μs	dlouhý =150μs krátký =73μs	79μs	dlouhý=150μs krátký= 73μs	79μs

$$1 \cdot SIFS + 2 \cdot Slot\ Time\ (AIFSN = 2) = 1 \cdot 16 \cdot 10^{-6} + 2 \cdot 9 \cdot 10^{-6} s = 34 \mu s.$$

ROVNICE 3.1 - PŘÍKLAD VÝPOČTU DOBY TRVÁNÍ PŘÍSTUPU K MÉDIU PRO UDĚLENOU PRIORITU HLASU

Tabulka 3.3 znázorňuje porovnání dob mezirámcových mezer DIFS, PIFS, SIFS a hodnoty CW_{min} a CW_{max} pro nejpoužívanější standardy 802.11.

TABULKA 3.3 - POROVNÁNÍ MEZIRÁMCOVÝCH MEZER A CW_{min} , CW_{max}

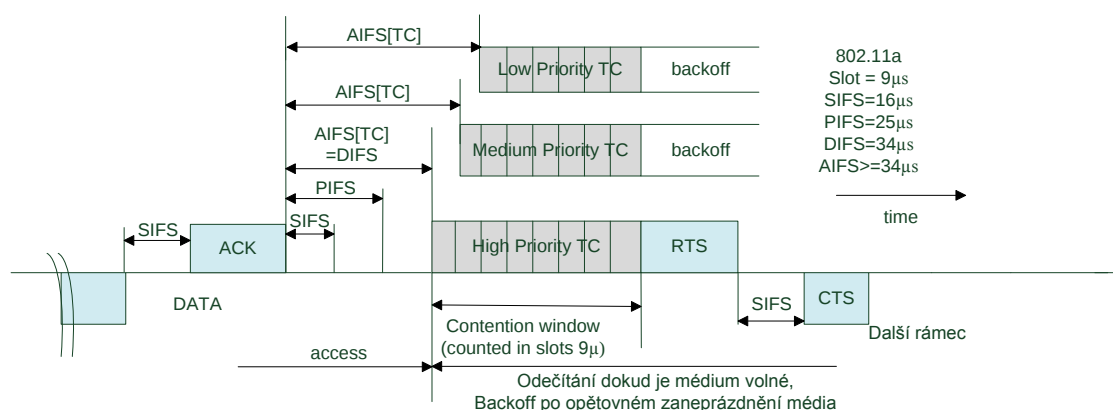
standard IEEE	DIFS [μ]	PIFS [μ]	SIFS [μ]	Cwmin	Cwmax
802.11a	34	25	16	15	1023
802.11b	50	30	10	31	1023
802.11g	50	30	10	15	1023
802.11n	2,4GHz krátký = 28 dlouhý = 50 5GHz - 34	2,4GHz krátký = 9 dlouhý = 20 5GHz - 25	2,4GHz - 10 5GHz - 16	15	1023

3.3 EDCF & HCF

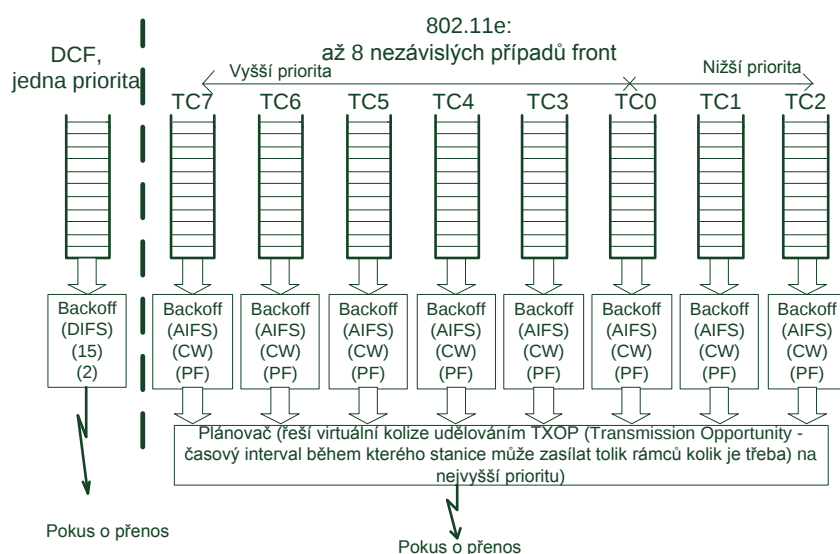
U standardu 802.11e nadále existují dvě fáze v rámci. Tyto dvě fáze CP a CFP se nepřetržitě střídají v čase. EDCF je používán pouze v CP, zatímco HCF je používáno v obou fázích CP a CFP, což činí z HCF hybridní koordinační funkci.

3.3.1 Enhanced Distributed Coordination Function (EDCF)

U rozšířené distribuované koordinační funkce jsou jednotlivé datové toky rozděleny do kategorií provozu (TC – traffic categories). EDCF používá různé mechanismy, které poskytují možnost rozlišení datového toku. Minimální okno sváru (CW_{min}) v backoff counteru je rozdílné pro rozdílné třídy priorit. To znamená, že přenosy u tříd s vyšší prioritou zaberou více času než přenosy tříd s nižší prioritou. Každá TC má také vlastní CW_{min} , CW_{max} a PF (persistence factor), které odlišují pravděpodobnost vítězství připojení k médium. PF určuje stupeň zvýšení hodnoty CW. TC s vyšší prioritou má nižší hodnotu PF. Další různé úseky rámce, mohou být použity pro rozdílné třídy priorit. EDCF je funkční pouze během CP. V průběhu CP, každá z traffic categories nezávisle uplatňuje ve stanicích možnost přenosu (TXOP – transmission opportunity). Všechny TC spouštějí po detekování volného kanálu backoff counter, pro uvolnění časového intervalu rovného AIFS. Hodnota AIFS je závislá na kategorii provozu, do které přenos spadá. EDCF je také základem pro funkci HCF [4].



OBRÁZEK 3.4 - PŘÍSTUP K MÉDIU EDCF



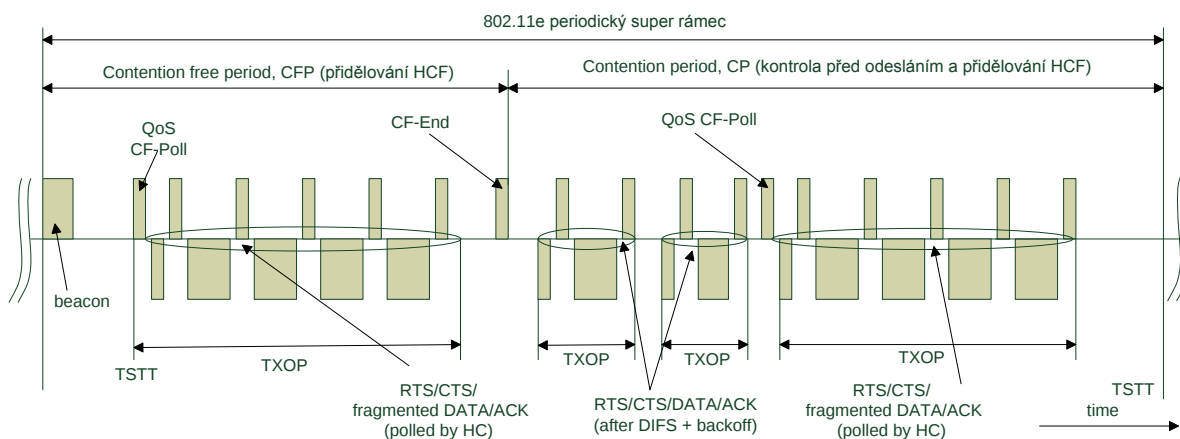
OBRÁZEK 3.5 - OSM PŘENOSOVÝCH KATEGORIÍ (TC), VLEVO DCF AIFS=34US CWMIN=15, PF=2; VPRAVO EDCF AIFS[TC]≥34US, CWMIN[TC]=0-255, PF[TC]=1-16 [8].

3.3.2 Hybrid coordination function (HCF)

Metoda HCF rozšiřuje pravidla přístupu EDCF. HC (hybrid coordinator) může přidělit TXOP z iniciativy MSDU (MAC service data unit). Doručení dat je možné kdykoliv, ale až po detekování volného kanálu po dobu PIFS, který je kratší než DIFS. Pro přidělení vyšší priority HC než EDCF, musí být hodnota AIFS delší než PIFS, tudíž musí být tato hodnota kratší než DIFS.

Během CP každá příležitost přenosu TXOP začíná v okamžiku, kdy je médium určeno k přenosu dle pravidel EDCF, tzn. po uplynutí AIFS plus backoff time, nebo když stanice přijme speciální poll-frame, QoS CF-Poll ze stanice HC. Rámec QoS CF-Poll může být odeslán z HC po uplynutí PIFS bez jakéhokoliv odstupu. Z tohoto důvodu může HC zažádat o TXOP v contention period pomocí jeho prioritního přístupu k médiu.

V průběhu CFP, startovací čas a maximální trvání každého TXOP je specifikováno stanicí HC, opět využívající QoS CF-Poll rámce. Stanice se nepokoušejí získat přístup k médiu z vlastní iniciativy během CFP, pouze HC může garantovat TXOP zasíláním QoS CF-Poll rámců. Doba CFP končí po uplynutí doby, která je uvedena v rámci beacon, nebo rámcem CF-End zaslaným z HC [4].



OBRÁZEK 3.6 - TYPICKÝ SUPER RÁMEC U 802.11E [9]

Řízené spojení během CP, je způsob pro HC, jak naučit stanice se ozvat, aby byly označeny, ve kterých časech a na jak dlouhou dobu jsou přístupné. Mechanismus kontrolovaného spojení umožňuje stanicím zasílat žádosti o přidělení označených TXOP, zasíláním zdrojů žádosti, aniž by se potýkaly s jiným přenosem (EDCF).

HC: zasílá kontrolní rámce pro kontrolované spojení.

Dědičné stanice: se neozývají v době intervalu kontrolovaného spojení.

Stanice zasílají žádost zdrojového rámce s požadovanou TC a dobou trvání TXOP.

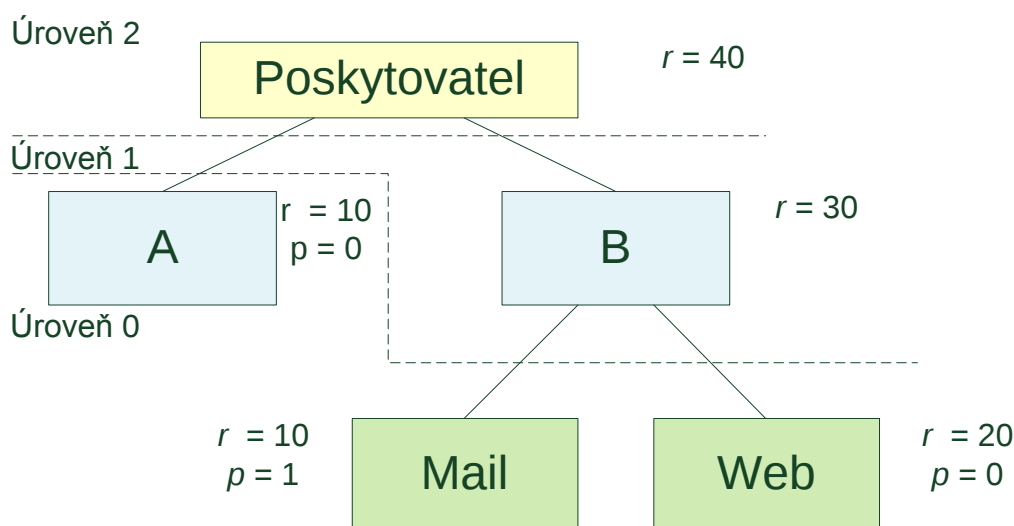
HC zasílá potvrzovací rámec ACK pro každou žádost zdroje.

4. Podpora QoS v operačních systémech Unixového typu

První část práce je zaměřena na popis QoS a na hardwarové řešení podpory QoS v bezdrátových sítích, které je zahrnuto ve standardu IEEE 802.11e. Následující kapitoly se budou zabírat různými mechanismy, které způsobem podpory QoS napodobují hardwarové řešení 802.11e. Budou popsány různé metody využívané v operačních systémech Linux, sdílení média, prvky využívané pro řízení datového toku a samotné komponenty, které jsou pro řízení datového toku potřebné. Linux nabízí bohatou sadu nástrojů pro správu a manipulaci s přenosem datových paketů.

4.1 Hierarchické sdílení média

Problém, kterým se zabývá každý poskytovatel, je rozdělení internetového pásma mezi dva nebo více uživatelů. Každý z těchto uživatelů má své specifické požadavky na datový tok a maximální odezvu, které chce mít garantované poskytovatelem. Velikost odezvy je závislá na přidělené prioritě datovému toku. Úkolem QoS je určení paketu, který má být odeslán jako další v pořadí, tzn., pakety mohou být odeslány ze všech stanic uživatelů na jednou, ale v případě, že některý z nich má udělenou prioritu přenosu, bude tento paket odeslán jako první. Na obrázku 4.1 je znázorněn příklad pro sdílení kapacity linky rozdělené pro dva účastníky, z toho jeden z nich má na poskytovatele jiné požadavky. Ve schématu jsou znázorněni Poskytovatel, účastníci A a B, přičemž B má požadavek pro rozdělení služeb na mail a web. Datový tok je označen r a priorita třídy p [10].



OBRÁZEK 4.1 - HIERARCHICKÉ SDÍLENÍ PŘENOSOVÉ KAPACITY

Pro pochopení a seznámení s podporou QoS je nutné se seznámit se základními termíny, které jsou využívány při probírání této problematiky.

- **třída** je uzel sdílejšího stromu s definovaným požadavkem na datový tok r .
- **uzel** je to pouze jiný název pro třídu.
- **list** je uzel bez dalších potomků. Pouze list může obsahovat pakety ve své interní frontě.
- **vnitřní uzel** je uzel, který není listem, tudíž má nějaké potomky.
- **plný list** je list obsahující nějaké pakety.
- **plný vnitřní uzel** je uzel s potomkem typu plný list.
- **podlimitní uzel** je takový uzel, který má aktuální tok nižší než je jeho požadovaný tok r .
- **nespokojený uzel** je současně plný uzel i podlimitní uzel – tento uzel by měl být obsloužen.
- **úroveň uzlu** je hloubka uzlu definovaná pro list jako 0. Číslování lze vyčíst z obrázku 4.1 [10].

4.2 Řízení datového toku

Řízení datového toku, je název, který je přiřazen systémům zabývajícím se řízením front a mechanismy obsluhující přijímání a vysílání paketů routeru. To zahrnuje rozhodování o tom, které pakety přijímat a zda je vůbec přijímat, v jakém poměru je přijímat a také jaké pořadí s jakým poměrem vysílání paketu bude probíhat na výstupu z rozhraní. V naprosté většině případů, se řízení datového toku skládá z jedné fronty, která sbírá přijímané pakety a řadí je do fronty tak rychle jak jen to hardware povolí. Tento druh fronty se nazývá FIFO (First In First Out). Výchozí queuing discipline (qdisc), což je algoritmus pro zpracovávání front, se v systémech Unixového typu nazývá pfifo_fast. Řízení datového toku je sada nástrojů umožňující kontrolu nad těmito frontami, a řídicích mechanismů poskytujících možnost obsluhy síťových zařízení. Pravomoc paketů, kterým byla přidělena priorita těmito mechanismy je vysoká, nenahradí ovšem odpovídající šířku pásma. Pro řízení datového toku se užívá zkratka QoS (Quality of Services). Pro ovládání datového toku se využívají následující prvky shaping, scheduling, classifying, policing, dropping, marking [11].

4.2.1 Shaping (tvarování)

Tvarování paketů je mechanismus, kterým se zpožďují pakety před přenosem do výstupní fronty, aby bylo dosaženo požadované výstupní rychlosti. Toto je jedním z nejčastějších přání uživatelů, kteří vyhledávají řešení pro řízení šířky pásma. Zpoždění paketu jako části řešení řízení datového toku, provádí každý tvarovací mechanismus. Pro zpoždění paketu je využíván mechanismus, který nevyužívá šetření práce pro vykonání zpoždění.

V opačném zobrazení to znamená, že mechanismus fronty nešetřící práci provádí tvarovací funkci. Mechanismus fronty šetřící práci by neměl být schopen zpoždění paketu.

Tvarovače se snaží omezit nebo přidělit provoz přenosového pásma, ovšem tak aby nepřesáhl předem definované limity uváděny v „packet per second“ (pps), „bit per second“ (bps) nebo „Byte per second“ (Bps). Jako vedlejší efekt mohou tvarovače vyhladit nárazový datový tok. Jednou z výhod tvarování šířky pásma, je možnost kontrolovat zpoždění paketů. Základním principem mechanismu pro tvarování je mechanismus známý pod názvem „token and bucket“. Jednoduše lze metodu Token bucket popsat jako algoritmus založený na analogii kbelíku s pevnou kapacitou, ve kterém jsou tokeny běžně reprezentovány jednotkami bytů nebo jednotlivými pakety o předdefinované velikosti, které jsou vkládány ve fixních časových intervalech. Pokud je třeba zkontrolovat pakety, zda vyhovují požadovaným limitům, tak kbelík kontroluje množství tokenů v daném čase. Pokud tokeny vyhovují, např. odpovídají délce paketu v bytech, tak se odstraní a paket je předán dál, např. k přenosu. Pokud v kbelíku není dostatek tokenů, paket neodpovídá a obsah kbelíku se nemění [11].

4.2.2 Scheduling (plánování)

Plánování je mechanismus pro uspořádání nebo přeuspořádání paketů mezi vstupem a výstupem konkrétní fronty. Nejčastěji používaný plánovač je plánovač FIFO. S větší perspektivou, může být nastavení mechanismu datového toku na výstupní frontu považováno za plánovač, protože pakety jsou uspořádány pro výstup. Ostatní plánovací mechanismy se pokouší kompenzovat různé síťové podmínky. Algoritmy pro spravedlivé řešení front (fair queuing algorithm SFQ) se pokouší zabránit dominanci jednoho klienta nebo datového toku v užívané síti. Algoritmus round-robin (WRR) přiřazuje každému datovému toku nebo spojení vlastní frontu na rozhraní síťové karty. Další plánovací algoritmy se zabývají zabráněním přetížení páteřní sítě (GRED) a zdokonalením dalších plánovacích algoritmů (ESFQ)[11].

4.2.3 Classifying (klasifikace)

Klasifikátory třídí nebo rozdělují datové toky do front. Klasifikace je mechanismus, kterým jsou pakety třízeny k různému zpracování nebo rozděleny do různých výstupních front. Během procesu schvalování, směrování a odesílání paketu může síťové zařízení klasifikovat paket řadou různých způsobů. Klasifikace může zahrnovat označení paketu, což se obvykle děje na hranici sítě pod jedním správcem nebo se může klasifikace objevit na každém skoku individuálně. Model Linuxu povoluje

cestu paketu kaskádovým stylem skrz několik klasifikačních skupin ve struktuře řízeného datového toku, přičemž může být klasifikován ve spojení s policery [11].

4.2.4 Policing

Policery opatřují měření a omezení určitého datového toku v určité frontě. Policing, jako prvek je jednoduchý mechanismus pro nastavení datového toku, kterým může být limitován datový přenos. Policing je nejčastěji používán na hranici sítě, aby zajistil, že spojení nebylo více náročné než přidělená šířka pásma. Policer akceptuje datový tok do určité úrovně, jakmile je tato úroveň překonána policer provádí akci pro provoz přesahujících rámců. Vcelku tvrdým řešením je shození provozu, i když by bylo možné provoz překlasifikovat namísto shození tohoto provozu. Policer klade otázku, zda provoz vstupující do fronty splňuje či nesplňuje podmínku provozu. Pokud se paket chystá vstoupit do fronty s povolenou rychlostí, policer provede akci a paket se do fronty zařadí. Pokud ale paket přichází s vyšší požadovanou přenosovou rychlostí, tak je provedena jiná akce. Ačkoliv policer používá vnitřně mechanismus token and bucket, nemá možnost zpomalit paket jako mechanismus tvarování [11].

4.2.5 Dropping

Dropping zahazuje celý paket, datový tok nebo klasifikaci. Dropping paketu je mechanismus, který celý paket zahodí. Pakety mohou být zahozeny z důvodů zahlcení sítě, poškození paketů během přenosu, plná vyrovnávací paměť nebo zahození způsobené nízkou hodnotou přiřazené priority provozu.

4.2.6 Marking

Marking je mechanismus používaný pro změnění paketů. Mechanismus marking u kontroly datového přenosu instaluje DSCP na samotné pakety, které jsou potom dále používány a respektovány ostatními routery uvnitř domény.

Každému z prvků pro řízení datového toku v OS Linux odpovídá některý z komponentů pro řízení QoS. Pro prvek shaping je to komponent class, pro scheduling qdisk. Další možnosti budou popsány v následujících kapitolách pojednávajících o jednotlivých komponentech [11].

4.2.7 QDisc

Qdisc - Queueing Discipline je algoritmus zpracovávající příchozí nebo odchozí frontu zařízení. Jednoduše řečeno je komponent qdisc plánovač. Každé výstupní rozhraní potřebuje plánovač nějakého druhu a výchozí plánovač jako je FIFO. Další přístupný qdisc v linuxu přeuspořádává pakety vstupující do plánované fronty v souladu s pravidly plánovače. Qdisc je hlavním stavebním blokem podpory QoS v operačních systémech Unixového typu.

Komponent qdisc se dělí na qdisc classfull a classless. Metoda classfull obsahuje a poskytuje možnost pro připojení filtrů, ve které není zakázáno používání třídy bez třídy s potomky. I když tato funkce bude většinou využívána a zpracovávána jinými systémy bez jakýchkoliv výhod.

Metoda classless qdisc nemusí obsahovat žádné třídy, k této metodě není ani možné připojit filtr. Vzhledem k tomu, že metoda classless neobsahuje žádné potomky, není zde zahrnut ani nástroj pro klasifikaci. To znamená, že žádný filtr nemůže být připojen ke classless qdisc.

Zdrojem pro zmatení terminologie qdisc je používání termínů root qdisc a ingress qdisc. Tyto pojmy nejsou skutečnými algoritmy pro zpracování front, ale spíš místa, ke kterým mohou být struktury podpory datového toku připojeny. Každé rozhraní obsahuje oba algoritmy root qdisc je pro příchozí pakety a ingress qdisc pro odchozí pakety [11].

4.2.8 Class

Třídy nabízí možnosti tvarovače. Třídy existují pouze u classfull qdisc (např. HTB a CBQ). Třídy jsou nesmírně pružné a mohou obsahovat jednoho nebo více potomků. Neexistuje žádný zákaz třídy zakazující obsahování samotné classfull qdisc, který usnadňuje řízení datového toku složitých scénářů.

Ke každé třídě může být připojeno libovolné množství filtrů, které povolují potomky nebo se používají pro přeuspořádání nebo zahození datového toku vstupujícího na konkrétní třídu.

Třída list je terminální třídou v qdisc. Ta zahrnuje výchozí FIFO a nikdy neobsahuje třídu potomka. Každá třída obsahující třídu potomka je root class a není třídou list.

4.2.9 Filter

Filtr je nejkomplexnějším komponentem pro QoS v operačních systémech typu Linux. Filtr poskytuje mechanismus pro spojení několika klíčových prvků datového provozu. Nejjednodušší a nejobvyklejší úlohou filtrů je klasifikace paketu. Linuxové klasifikátory nemohou existovat mimo

filtr. Filtry linuxu umožňují uživateli zařadit pakety do výstupní fronty buď s použitím několika filtrů, nebo s použitím pouze jednoho filtru.

Filtr musí obsahovat klasifikátor, ale nemusí obsahovat policer.

Filtry mohou být spojeny jak s classfull qdisc tak i s classless qdisc., nicméně paket nezařazený do fronty vždycky nejprve vstupuje do root qdisc. Potom co se filtr připojí k root qdisc, může být paket přesměrován do různých podtříd, ve kterých může podstoupit další klasifikaci [11].

4.2.10 Classifier

Objekty filtru, kterými může být manipulováno pomocí nástroje tc (traffic control), mohou používat několik rozdílných klasifikačních mechanismů, z nichž je nejobvyklejší klasifikátor u32. Tento klasifikátor umožňuje uživateli třídit pakety na základě jejich vlastností. Klasifikace je také možná pomocí nástroje netfilter, který je obsažen v linuxovém jádře a je ovládán pomocí iptables.

Klasifikátory jsou nástroje, které mohou být použity jako části filtru pro identifikaci charakteristiky paketů nebo paketových metadat (data pro řízení paketu). Objekt klasifikátor je v Linuxu přímou analogií pro základní operaci a mechanismus klasifikace řízení datového toku.

4.2.11 Policer

Tento základní mechanismus je používán pouze u řízení datového toku v Linuxu jako část filtrů. Policer vyvolává jednu akci, když dojde k překročení limitu datového toku a jinou akci když je rychlost toku pod limit. I když jsou oba základní elementy Policing a Shaping založeny na limitování šířky pásma, tak policer nebude nikdy zpožďovat přenos. Může provést pouze akci založenou na základě požadovaných kritérií.

4.2.12 Drop

Tento základní mechanismus pro řízení QoS je používán pouze v Linuxu jako část policeru. Jakýkoliv policer se připojí k jakémukoliv filtru a může vykonat akci zahození paketu. Existují však místa v systému kontroly datového toku, kde mohou být data zahozena z důvodu postranního efektu. Například může být paket zahozen, pokud aktivní plánovač používá metodu kontroly toku, jako to dělá systém GRED. Také tvarovač nebo plánovač, který běží mimo svou přidělenou vyrovnávací paměť, mohou zahodit paket během nárazového nebo přetíženého období [11].

4.2.13 Handle

Každá třída a classfull qdisc vyžadují jedinečný identifikátor v rámci struktury řízení datového toku. Tento jedinečný identifikátor je známý jako handle a skládá se ze dvou členů, majoritní číslo a minoritní číslo. Tato čísla lze přiřadit libovolně uživatelem v souladu s následujícími pravidly.

Číslování handle pro třídu a qdisc:

- Majoritní číslice – Tento parametr nemá s jádrem žádnou spojitost. Uživatel může použít libovolné schéma číslování, nicméně všechny objekty v struktuře kontroly datového toku se stejným rodičem, musí sdílet majoritní číslo handle.
- Minoritní číslice – tento parametr jednoznačně identifikuje objekt jako qdisc, pokud je hodnota 0. Každá jiná hodnota identifikuje objekt jako třídu. Všechny třídy sdílející rodiče musí mít jedinečné minoritní číslo.

Speciální hodnota handle je ffff:0, která je rezervována pro root qdisc.

Handle je použit jako cíl pro ID třídy a ID toku pro příkazy tc. Tento typ handle je externí identifikátor objektu, použitelný pro uživatelské aplikace. Jádro udržuje vnitřní identifikátory pro každý objekt [11].

5. Fronty

Pomocí front se určuje, v jakém pořadí jsou data odesílána, ale nemáme žádnou kontrolu nad tím, v jakém pořadí jsou přijímána. Pakety využívají k odesílání pomalého startu, což znamená, že jsou nejprve odesílány pomalu a rychlost odesílání se postupně zvyšuje do té doby, dokud nejsou pakety odmítnuty. Tím je možno kontrolovat jaký datový tok je linka schopna pojmout, před tím než data dorazí do routeru. To znamená, že bychom měli nastavit queuing discipline qdisc v routeru, pro přeskokování paketů dle našich pravidel pro všechna síťová zařízení připojená k místní síti ne k internetu.

Každé síťové zařízení má možnost nastavení qdisc. Jako výchozí bývá nastaveno pfifo_fast, čím se bude zabývat jedna z následujících kapitol.

V operačních systémech typu Linux jsou dva různé typy front. Prvním typem je Classless, který neumožňuje přiřazovat další qdisc k zařízením. Druhým typem je Classful umožňující vytvoření tříd, které pracují jako větve stromu. Poté je možné nastavit filtrování paketů každé z tříd. Každá z tříd může mít přiřazeno další classless nebo classfull qdisc [11].

5.1 Classless

Classless fronty, jsou fronty, které řídí základní typy provozu pomocí přeuspořádání, zpomalení a zahození paketu. Tento typ qdisc neumožňuje vytvoření tříd.

5.1.1 FIFO

FIFO algoritmus tvoří základ pro výchozí disciplíny qdisc ve všech síťových rozhraních Linux (pfifo_fast). Neprovádí žádné tvarování nebo přeskupování paketů. Jednoduše vysílá pakety, hned jakmile je to možné po jejich přijetí a zařazení. Je to také qdisc používaný uvnitř nově vytvořených tříd jinými typy qdisc nebo třída nahrazující FIFO.

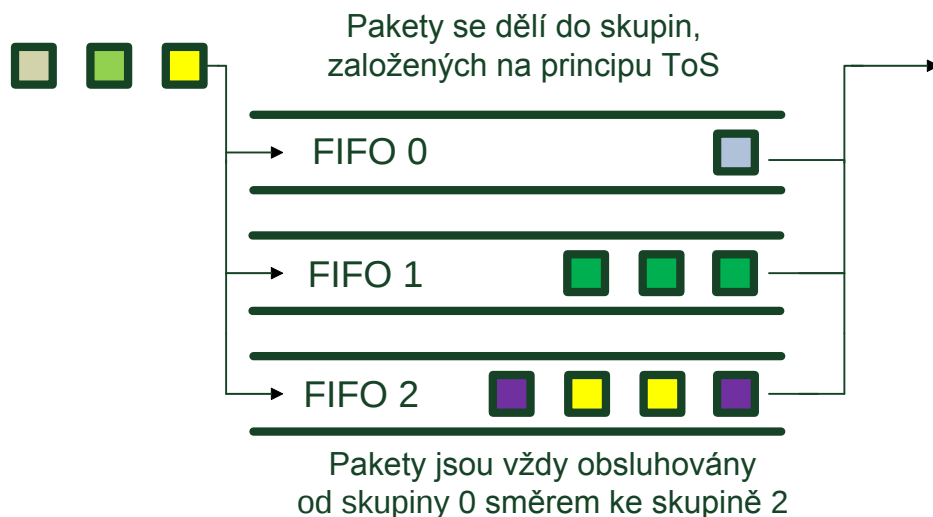


OBRÁZEK 5.1 – ZÁKLADNÍ ŘAZENÍ DO FRONT FIFO

QDisc disciplína FIFO musí mít nějaký limit (velikost vyrovnávací paměti) pro předcházení přetékání v případě, že FIFO nestíhá pakety řadit do fronty tak rychle jak je přijímá. Linux implementuje dva základní typy FIFO. Jeden je založený na bytech bfifo a druhý je založený na paketech pfifo. Bez ohledu na používaný typ FIFO je velikost fronty definována limitním parametrem [11].

5.1.2 pfifo_fast

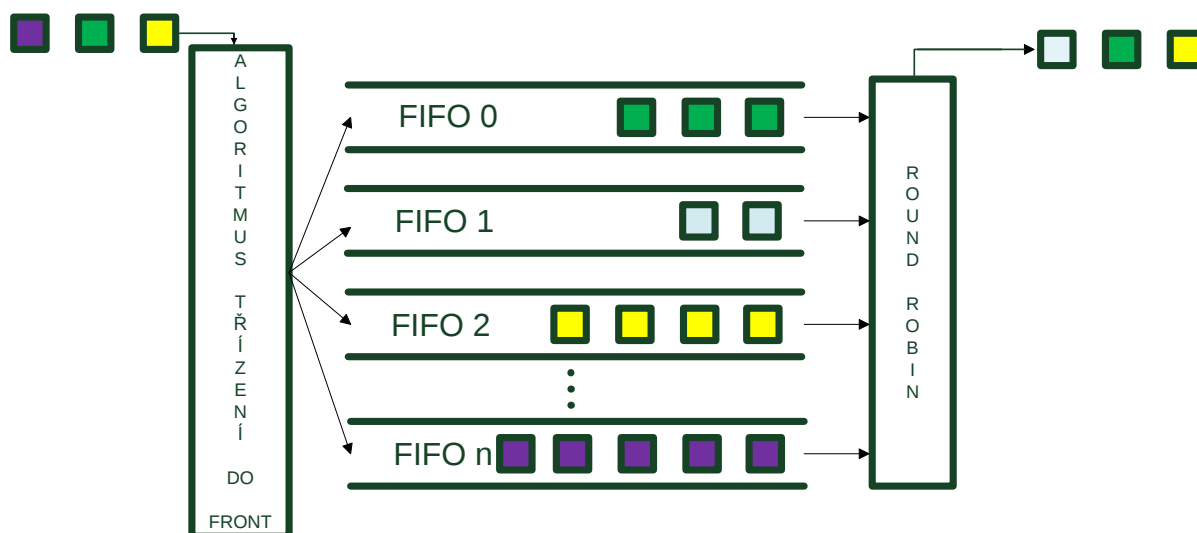
Pfifo_fast je výchozí qdisc pro všechna rozhraní v Linuxu. Je založen na běžném FIFO qdisc, ale poskytuje také určitou prioritizaci. Nabízí tři různá pásma (individuální FIFO) pro rozdělení datového provozu. Nejvyšší priorita provozu (interaktivní datové toky) je umístěna v pásmu 0 a je vždycky obsloužena jako první. Podobně pásmo 1 je vždy obslouženo dříve, než pásmo 2 viz obrázek 5.2. Koncový uživatelé, nemohou u disciplíny pfifo_fast nic konfigurovat, tato disciplína je výchozí.



OBRÁZEK 5.2 – VÝCHOZÍ QDISC V OS UNIXOVÉHO TYPU PFIFO_FAST

5.1.3 SFQ

SFQ – Stochastic Fair Queueing se pokouší spravedlivě rozdělovat možnosti pro přenos dat do sítě mezi libovolnými počty datových toků. Dosahuje toho pomocí funkce hash, která dělí datové toky na jednotlivé vnitřně ovládané FIFO, které jsou řazeny do fronty metodou cyklické obsluhy (Round – Robin). Vzhledem k tomu, že je možnost nespravedlnosti při výběru funkce hash, je tato funkce periodicky měněna. Tato periodicitu je nastavena pomocí odchylky.



OBRÁZEK 5.3 – SFQ METODA PRO SPRAVEDLIVÉ DĚLENÍ DAT DO FRONT

Naneštěstí, některé druhy softwaru (např. Kazaa a eMule), ničí výhodu tohoto pokusu o spravedlivé fronty tím, že spouští tolik TCP datových toků, kolik je jen možné. V mnoha sítích se slušnými

uživateli, může SFQ distribuovat síťové zdroje pro soupeření s jinými datovými toky a další opatření mohou být zahájena, po napadení sítě [11].

5.1.4 ESFQ

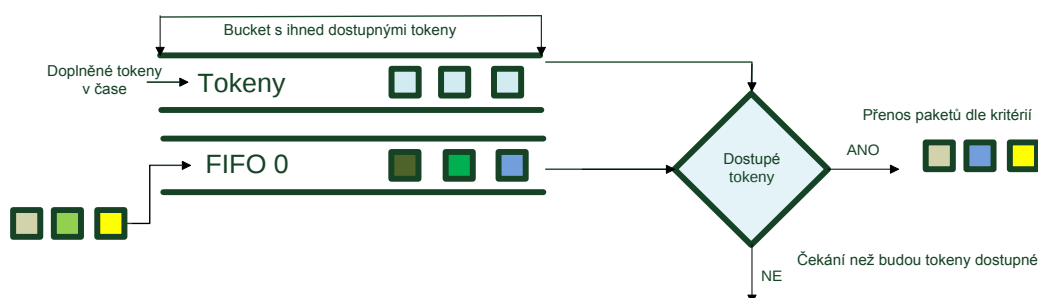
ESFQ – Extended Stochastic Fair Queueing není konceptem jinou disciplínou než samotné ESF. Povoluje pouze uživateli kontrolovat více parametrů než jednodušší předchůdce. ESFQ byl koncipován tak, aby předcházel nedostatkům, které jsou v jednoduchém SFQ. Povoláním kontroly hash algoritmu pro uživatele, se dosáhlo větší kontroly šířky pásma, čímž je pro uživatele uskutečněno dosažení spravedlivějšího rozdělení šířky pásma.

5.1.5 GRED

GRED – Generic Random Early Drop je nastavba algoritmu RED, který jednoduše pomocí algoritmu v routeru nebo v jiném síťovém zařízení, pokud je zaplněna vyrovnávací paměť, zahazuje některé z paketů. Pokud bude vyrovnávací paměť neustále plná tak dochází k přetížení sítě. Metoda GRED je užitečná pro páteřní či hlavní sítě, není vhodná pro koncové uživatele [11].

5.1.6 TBF

TBF – Token Bucket Filter -tento qdisc je založen na metodě Token Bucket viz kap. 4.2.1. Jednoduše tvaruje datový tok přenášený na rozhraní. Pro omezení rychlosti, při kterém budou pakety řazeny do front dle určitého rozhraní, je TBF ideální řešení. Snadně zpomaluje datový tok na předepsanou rychlost. Pakety jsou přenášeny pouze tehdy, existuje-li dostatečné množství tokenů. V opačném případě jsou pakety časově rozlišeny. Odložené pakety jsou tímto způsobem uměle zpožděny o určitý čas [11].



OBRÁZEK 5.4 – METODA PRO TVAROVÁNÍ DATOVÉHO TOKU TOKEN BUCKET FILTER - TBF

5.2 Classful

Třídy typu classful mohou obsahovat další classless i classful qdisc. Tímto tvoří strom tříd skládaný z uzlů obsahujících další classful qdisc a listů obsahujících jeden classless qdisc. Každé síťové rozhraní má na výstupu jednu kořenovou disciplínu qdisc nastavenou do výchozího nastavení pfifo_fast, která je používána pro tvarování odchozího směru. O vstup na rozhraní se stará policing, který má pouze omezené možnosti nastavení.

5.2.1 PRIO

Qdisc PRIO ve skutečnosti netvaruje datový tok, pouze rozděluje data v závislosti na tom, jak jsou nastaveny filtry. PRIO lze považovat za druh pfifo_fast „na steroidech“, přičemž každá skupina je oddělená třída namísto jednoduchých FIFO.

Pokud je paket přiřazen k disciplíně PRIO, třída je vybrána na základě příkazů filtrů, které byly nastaveny. Ve výchozím nastavení jsou vytvořeny tři třídy. Tyto třídy ve výchozím nastavení obsahují čisté disciplíny FIFO bez vnitřní struktury, ale je možné je zaměnit jinými qdisc, které jsou dostupné.

Kdykoliv potřebuje být paket zařazen do třídy 1 je to vykonáno jako první. Vyšší třídy jsou použity pouze tehdy, pokud se všechna nižší pásma vzdala paketu.

Disciplína PRIO je velmi užitečná v případě, že je třeba upřednostnit určité druhy přenosu bez používání ToS – značky, ale za použití všech možností filtrů tc. Je také možné přiřadit další qdisc do tří předem předdefinovaných tříd, přičemž pfifo_fast je limitováno jako jednoduchý qdisc FIFO.

Protože PRIO ve skutečnosti netvaruje datový tok, platí zde stejné omezení jako u SFQ. Je možné jej použít pouze, pokud je fyzické spojení plné nebo je PRIO zahrnuto uvnitř třídy, která přenos tvaruje. Toto pravidlo platí téměř u všech kabelových modemů a zařízení DSL [12].

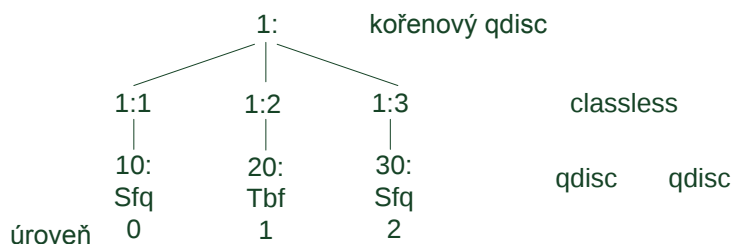
5.2.1.1 Parametry PRIO a užití

Kategorie provozu rozlišují u qdisc PRIO následující parametry:

- Úrovně – počet úrovní, které je třeba vytvořit. Každá úroveň je ve skutečnosti třída. Pokud je změněn počet úrovní je třeba změnit také:
 - o Priomap – pokud nejsou poskytnuty kategorie provozu pro klasifikaci provozu, qdisc PRIO sleduje TC-PRIO pro rozhodnutí jak rozdělit provoz.

Pracuje to stejně jako qdisc pfifo_fast popsán v kapitole 5.1.2.

Úrovně jsou třídy a mají výchozí název: hlavní 1, hlavní 2, hlavní 3. Pokud qdisc PRIO je nazván např. 12, tak mu filtr kategorie provozu poskytne vyšší prioritu [12].



OBRÁZEK 5.5 - PŘÍKLAD PRIO

5.2.2 CBQ

CBQ – Class Based Queueing – je nejkomplexnější dostupná qdisc, nejvíce medializovaná a pravděpodobně nejkomplikovanější disciplína. CBQ algoritmus není přesný a neodpovídá tomu, jak Linux funguje.

CBQ není jenom o tom, že třídí pakety do tříd, ale také tvaruje datový tok a v tomto ohledu nepracuje příliš dobře. Pokud se tato disciplína pokusí vytvarovat přenos z 10 Mbit/s na 1 Mbit/s, měla by linka zůstat z 90% času volná. Pokud tomu tak není, je třeba ji přiškrtnout, aby tomu tak bylo.

Tuto dobu je docela složité změřit, tak CBQ odvozuje dobu nečinnosti z počtu mikrosekund, které uplynou od žádosti z hardwarové vrstvy pro další data. Ve výsledku to může být používáno pro odhad jak plné nebo volné spojení je.

Je to poněkud složitý způsob a ne vždy přináší správné výsledky. Např. co když skutečná rychlost linky pro rozhraní není schopna přenést plných 100 Mbit/s dat, možná z důvodu použití nesprávného ovladače? Síťová karta PCMCIA nikdy nedosáhne 100 Mbit/s, protože je tak navržena sběrnice.

Může to být ještě horší, pokud budou uvaženy ne docela reálné síťové zařízení jako PPP pro ethernet nebo PPTP pro TCP/IP. Efektivní šířka pásma je zde určena účinností linek k uživateli, které jsou obrovské. Lidé, co prováděli měření, zjistili že CBQ není vždy velmi přesné a v mnoha případech je spíš zcela nepřesné. Nicméně v mnoha případech pracuje správně [12].

5.2.2.1 Tvarování (Shaping) u CBQ

CBQ pracuje tak, že se ujišťuje, je-li linka dostatečně dlouho volná pro uskutečnění přenosu dle nakonfigurované rychlosti v reálné šířce pásma. Vypočítává čas, který by měl projít mezi průměrnými pakety.

V průběhu samotné operace se efektivní doba nečinnosti měří pomocí exponenciálního klouzavého průměru (EWMA), který bere v úvahu nedávné pakety, které jsou exponenciálně důležitější než ty staré. Průměrné zatížení se v UNIXu vypočítává stejným způsobem.

Vypočtená doba klidového stavu se odečte od jedné změřené EWMA, výsledné číslo je nazváno „avgidle“ (průměrná nečinnost). Dokonalá hodnota avgidle je 0: pakety dorazí přesně jednou za vypočtený interval. Pokud je avgidle záporné a dosáhne vysoké záporné hodnoty, dojde k vypnutí CBQ na krátkou dobu a poté nastává přetížení linky. V opačném případě, avgidle s nečinností linky roste a po několika hodinách nečinnosti by mohla vzniknout nekonečná šířka pásma. Pro předejití tohoto problému je nastavena hodnota maxidle.

Pokud je linka přetížena, teoreticky by CBQ měl přiškrtit tok na dobu, která byla vypočítána pro projití mezi pakety a dále propuštění paketu a opětovné přiškrcení. Více je popsáno v následujících parametrech.

Parametry specifikované pro konfiguraci tvarování:

- Avpkt – průměrná velikost paketu, měřená v bytech. Je třeba pro výpočet maxidle, který je odvozen z maxburst specifikovaném v paketech.
- Bandwidth – fyzická šířka pásma zařízení je třeba pro výpočet doby nečinnosti.
- Cell (buňka) – doba, kterou potřebuje paket pro přenos zařízením. Může růst v krocích, což je závislé na velikosti paketu.
- Maxburst – tato hodnota se používá pro dopočítání maximální doby nečinnosti maxidle, takže v dobu, kdy avgidle je v hodnotě maxidle. Tato průměrná hodnota paketů se může změnit před tím než avgidle nabude hodnoty 0. Čím vyšší hodnota maxburst tím vyšší tolerance doby pro změnu. Není možné nastavit maximální dobu nečinnosti maxidle přímo, jen s použitím maxburst.
- Minburst – jak bylo zmíněno dříve CBQ přiškrcuje datový tok v případě nadlimitního datového toku. Ideální řešení, by bylo tak provést pro přesně vypočítanou dobu nečinnosti a propustit pouze jeden paket. Pro jádra UNIX je však obtížné naplánovat události kratší než 10ms, tudíž je lepší tok přiškrtit po delší dobu a poté propustit minburst paket v jednom čase a následně tento paket na delší dobu uspat. Čekací doba je nazývána offtime. Vyšší hodnota minburst vede k přesnějšímu tvarování v dlouhém čase, ale po větších dávkách v milisekundových lhůtách.
- Minidle – pokud je hodnota avgidle nižší než 0, přenos je nadlimitní a třeba počkat do doby než bude avgidle dostatečně velký pro odeslání paketu. Pro zabránění náhlého vypnutí linky na delší dobu, je avgidle nastaveno na minidle. Minidle se uvádí v záporných hodnotách, což znamená, že hodnota 10 je rovna pro omezení avgidle na -10.

- Mpu – minimum packet size – minimální velikost paketu je třeba, protože nulová velikost paketu je přiřazena 64 bytům u ethernetu a to zabere určité množství času. CBQ potřebuje tuto hodnotu vědět pro vypočítání délky doby nečinnosti.
- Rate – požadovaná rychlost přenosu opouštějícího tento qdisc.

CBQ má vysokou citlivost ladění. Např. třídy, které nemají data ve frontě, se o tuto disciplínu nezajímají. Nadlimitní třídy jsou penalizovány tím, že je snížena jejich efektivní priorita [12].

5.2.2.2 Chování CBQ classful

Kromě tvarování za použití výše popsaných aproximací, působí CBQ také jako řazení do front PRIO, v tom smyslu, že mohou mít odlišné priority a data s nižší prioritou budou tázána před daty s prioritou vyšší. Pokaždé, když je paket vyzván hardwarovou vrstvou k odeslání do sítě, startuje vážený round robin proces (weighted round robin WRR) s třídami s nižší prioritou. Třídy jsou potom seskupeny a dotazovány zda mají k dispozici data. Pokud ano třída je vrácena. Potom co je povoleno třídě zařadit počet bytů, snaží se o to další třída v rámci.

Parametry řízení procesu WRR:

- Allot – přidělování – pokud zažádá CBQ pro zaslání paketu na rozhraní, bude to vyzkoušeno všemi qdisc (ve třídách) v pořadí přidělených priorit. Pokaždé, kdy třída dostane pokus pro odeslání dat, může odeslat pouze omezené množství dat. Allot je základní jednotkou této hodnoty.
- PRIO – CBQ může také působit jako zařízení PRIO. Vnitřní třídy s vyšší prioritou se pokouší o odeslání dat jako první tak dlouho dokud mají nějaký datový tok, ostatní třídy nejsou tázány.
- Weight – hmotnost – pomáhá WRR procesu. Každá třída dostane šanci pro odeslání dat. Pokud existují třídy s výrazně větší šířkou pásma, tak jim je umožněno zaslat více dat v jednom kole než ostatní třídy. CBQ sčítá všechny hodnoty weight a normalizuje je, tudíž je možné jich použít libovolné množství.

5.2.2.3 CBQ parametry určující sdílení a zapůjčení linky

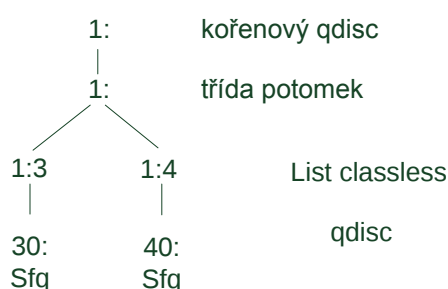
Kromě omezování na určité datové přenosy, lze specifikovat, které třídy si mohou půjčit kapacitu linky od jiné třídy nebo naopak poskytnout svou šířku pásma.

- Izolovaní/sdílení (isolated/sharing) – třídy, které jsou nakonfigurovány, jako izolované nebudou půjčovat své pásmo sourozeneckým třídám. To se používá, pokud jsou na lince

dva soupeřící subjekty, které si nechtějí poskytnout šířku pásma zdarma. Kontrolní program tc také zná sdílení, které je opakem izolování.

- Omezení/zapůjčení (bounded/borrow) – třída může být také omezená, což znamená, že se nebude pokoušet vypůjčit pásmo od sourozeneckých tříd. Zapůjčení je opakem omezení.

Typickou situací je, že jsou na lince dva subjekty a oba dva jsou izolované a omezené. To znamená, že jsou limitovány na jejich přiřazené rychlosti, a také si nemohou půjčovat pásmo navzájem. V takovém případě může být vložena další třída, které je umožněno přepínání šířky pásma [12].



OBRÁZEK 5.6 - PŘÍKLAD CBQ

5.2.3 HTB

HTB – Hierarchical Token Bucket – využívá konceptu token a bucket spojeného se systémem založeným na třídách a filtrech, pro povolení komplexní kontroly nad přenosem. S komplexním modelem půjčování, může HTB provádět celou řadu sofistikovaných technik kontroly přenosu. Jednou z nejjednodušších metod použití HTB je použití tvarování.

Pro pochopení metody HTB, je důležitým logickým krokem chápat metodu token bucket a funkci TBF. Tato disciplína front umožňuje uživateli definovat charakteristiku metody token a bucket a povoluje vkládat do bucketu tokeny svévolně. Při spojení metody se schématem klasifikace, může být přenos kontrolován velice podrobným způsobem.

5.2.3.1 Požadavky na software

HTB je novější disciplínou front, tím pádem distribuce Linux nemusí obsahovat všechny nástroje a kapacity, které jsou třeba. Pro povolení podpory HTB u starších verzí lze použít iproute2 - oprava pro tc.

5.2.3.2 Tvarování

Jedna z nejpoužívanějších aplikací u HTB zahrnuje tvarování přenášeného datového toku na specifickou rychlost. Všechna tvarování probíhají ve třídě listu. V příchozích datech nebo vnitřních uzlech nedochází ke tvarování, neboť existují pouze proto, aby určili, jak by měl model zapůjčení distribuovat dostupné tokeny [13].

5.2.3.3 Půjčování

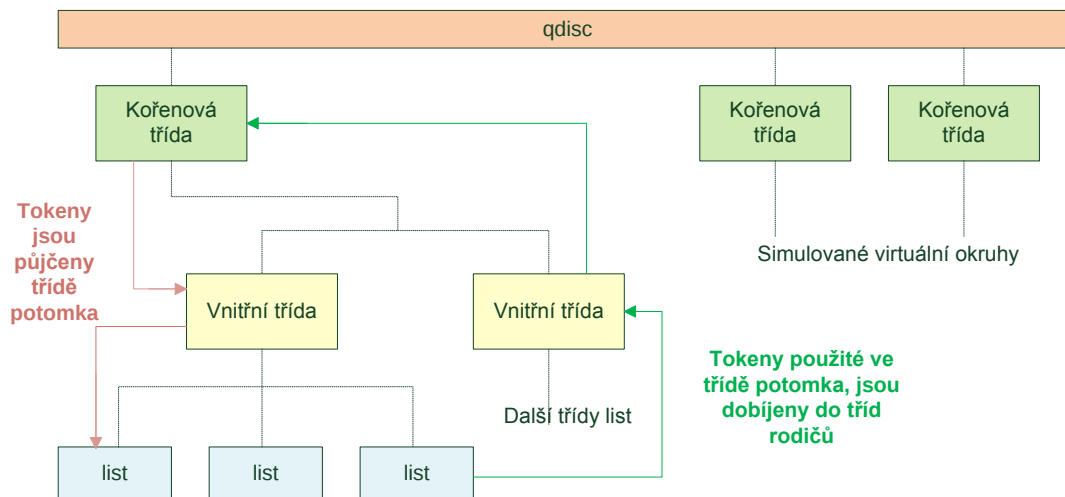
Základní částí qdisc HTB je mechanismus půjčování (borrowing). Třídy potomků si půjčují tokeny od tříd rodičů poté, co překročí jejich spodní limitní hranici (rate). Třída potomka se bude pokoušet o zapůjčení tokenů do doby, dokud nedosáhne bodu horní limitní hranice (ceil), od kterého může řadit pakety pro přenos, až do doby kdy budou k dispozici další tokeny. Jsou zde pouze dva základní typy tříd, které mohou být vytvořeny HTB. Následující tabulka 5.1 a digram viz obrázek 5.7 znázorňují rozdílné možné stavy a chování mechanismu půjčování.

TABULKA 5.1 - HTB – PŮJČOVÁNÍ TOKENŮ

typ třídy	stav třídy	vnitřní stav HTB	uskutečněná akce
list	< rate	HTB může posílat	třída list vyprazdňuje frontu až do doby dokud jsou k dispozici tokeny
list	> rate < ceil	HTB může půjčovat	třída list se bude snažit půjčovat tokeny od třídy rodiče.
list	> ceil	HTB nemůže posílat	žádné pakety nebudou zařazeny, což způsobí zpoždění paketu a tím vznikne požadované zpoždění
vnitřní kořen	< rate	HTB může posílat	vnitřní třída půjčuje tokeny třídě potomek
vnitřní kořen	> rate < ceil	HTB může půjčovat	vnitřní třída se pokouší půjčit tokeny od rodiče a poté je půjčuje potomkům
vnitřní kořen	> ceil	HTB nemůže posílat	vnitřní třída se nesnaží vypůjčit tokeny a ani je nepůjčuje potomkům

Následující diagram identifikuje tok půjčených tokenů a chování při vracení tokenů zpět třídě rodiče. Aby mohl model půjčování fungovat, musí mít každá třída přesný počet tokenů pro použití pro sebe a svoje potomky. Z tohoto důvodu, je každý token použit třídou potomka nebo listu a je přičítán třídě rodiče až do doby, kdy nedosáhne třídy kořene.

Každá třída potomka, která chce půjčit token, bude o token žádat třídu rodiče, který pokud nemá volný token, žádá třídu svého rodiče až do doby dokud není dosaženo třídy s volným tokenem nebo je dosaženo třídy kořene. Takže půjčování tokenů proudí směrem tříd list a nabíjení tokenů proudí směrem kořenových tříd [11].



OBRÁZEK 5.7 – PŘÍKLAD VYPŮJČOVÁNÍ TOKENŮ

5.2.3.4 Parametry třídy HTB

- Výchozí – parametry jsou volitelné u každého qdisc HTB objektu, výchozí nastavení je 0, což způsobuje každý neklasifikovaný provoz je zařazen pod hardwarovou rychlost, zcela obchází všechny připojené třídy kořenového qdisc.
- Rate – slouží pro nastavení minimální požadované přenosové rychlosti. To lze považovat za ekvivalent Committed Information Rate (CIR – požadovaná informační rychlost), nebo garantované šířky pásma pro danou třídu list.
- Ceil – slouží pro nastavení maximální požadované rychlosti přenosu. Model půjčování může ilustrovat, jak je tento parametr použit.
- Burst – velikost sazby bucketu (viz token a bucket). HTB bude zařazovat burst byty než dorazí další tokeny.
- Cburst – udává velikost bucketu ceil. HTB bude zařazovat cburst byty než dorazí další ctokeny.
- Quantum – množství – klíčový parametr používaný HTB pro kontrolu půjčování. Za normálních okolností je množství počítáno HTB, ne konkrétním uživatelem. Ladění tohoto parametru může mít obrovský vliv, protože se používá pro rozdělení provozu mezi třídami potomků a pro přenos paketů ze stejných tříd.
- r2q – také se obvykle počítá pro uživatele. R2q je náznak HTB pro pomoc určit optimální množství pro určitou třídu.

5.2.3.5 Pravidla

Níže jsou uvedeny základní pokyny pro používání HTB.

- Tvarování se u HTB vyskytuje pouze ve třídách list.
- Protože HTB netvaruje třídy pouze třídy typu list, součet minimálních přenosových rychlostí by neměl přesahovat maximální přenosovou rychlost rodičovské třídy.
- Pouze třída list tvaruje přenos. Pakety jsou v těchto třídách jenom zpoždovány. Vnitřní třídy existují pouze pro definování jak půjčovat/ vypůjčovat tokeny.
- Množství se používá pouze, pokud je třída nad minimální požadovanou přenosovou rychlost a pod maximální požadovanou přenosovou rychlost.
- Množství by mělo být nastaveno jako nejvyšší velikost paketu nebo ještě vyšší. HTB bude řadit jednotlivé pakety při nejmenším do doby, kdy množství nebude příliš malé. V takovém případě nebude možné vypočítat skutečnou šířku pásma.
- Třídy rodičů půjčují tokeny třídám potomků a zvyšují hodnotu množství.
- Rozdíl mezi tokeny a ctokeny má smysl pouze ve třídě listu, protože ostatní třídy pouze tokeny půjčují [11].

5.2.4 HFSC

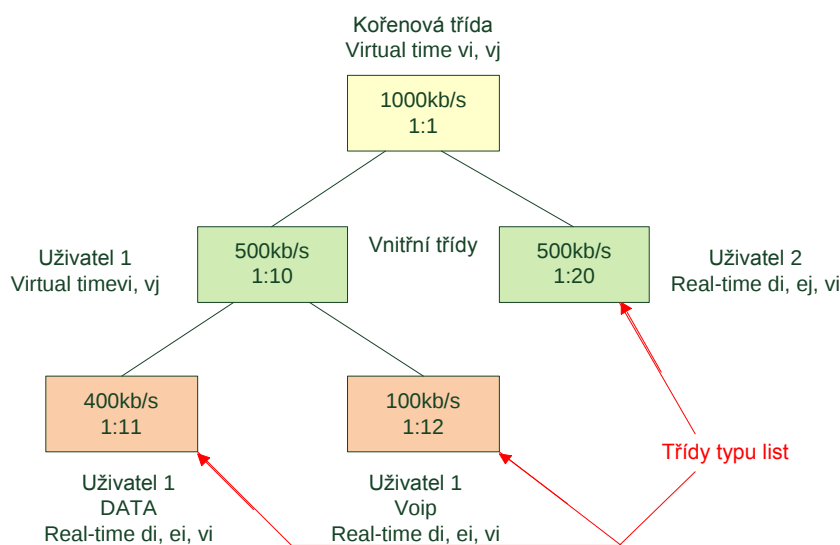
HFSC – Hierarchical Fair Service Curve se blíží servisní křivce modelu sdílení linky. Křivka zde definuje požadavky QoS třídy datového toku. Plánování je zde založeno na dvou kritériích: real-time kritéria, které zajišťují garanci služeb všech listových tříd, a kritéria sdílení linky (link-sharing), která si klade za cíl uspokojit servisní křivku vnitřních tříd a spravedlivě rozdělit přebytečnou šířku pásma. Kritéria real-time jsou používána k výběru paketů, pouze pokud hrozí potenciální nebezpečí, že garance služeb pro uzly typu list jsou v ohrožení. V opačném případě jsou využity kritéria sdílení linky. Tato politika zajišťuje garanci v reálném čase třídám typu list, zatímco ve stejném čase se minimalizuje rozdíl mezi aktuálními službami přijímanými do vnitřních uzlů a ty jsou řízeny modelem sdílení linky.

U HFSC má každá listová třída i přidělenou trojici parametrů (e_i, d_i, v_i) , zatímco každá interní třída j si udržuje pouze parametr (v_j) , kde e_i a d_i představují časovou náročnost a termín spojené s prvním paketem ve třídě fronty i , a v_i a v_j jsou virtuálními časy tříd. Termíny jsou přiřazeny tak, aby byly splněny všechny termíny v relaci, čímž se zaručuje servisní křivka. Časová náročnost se používá k určení, které ze dvou plánovacích kritérií použít pro výběr příštího paketu. Paket v čele fronty i spadá do časové náročnosti pokud $e_i \leq t$, kde t je aktuální čas. Časové náročnosti jsou počítány tak, aby v každém okamžiku, kdy jsou časově náročné pakety v systému, čímž vzniká nebezpečí vypršení

termínu, při používání kritéria sdílení linky namísto real-time kritéria, tj. vzniká potenciál konfliktu mezi cíly kritérií sdílení linky a real-time. Od té doby, kdy je důležitější se zaměřit na služby v reálném čase (real-time), se v případě výskytu časově náročných paketů vždy využívá kritérium real-time, který vybírá mezi všemi časově náročnými pakety jeden s nejkratším termínem. V každém čase, kdy nejsou dostupné časově náročné pakety, tj. není možnost konfliktu mezi kritérii sdílení linky a real-time, algoritmus bude aplikovat rekurzivně kritérium sdílení linky. Algoritmus bude začínat v kořenové třídě a končit ve třídě list, vybírá mezi všemi třídami potomka tu třídu, která má nejkratší virtuální čas. Zatímco termín a časová náročnost jsou spojeny pouze se třídami list, virtuální čas je spojen s oběma vnitřními třídami i se třídami typu list. Virtuální čas třídy představuje normalizovanou hodnotu služby, která byla přijata od třídy. V ideálně spravedlivém systému jsou všechny virtuální časy sourozeneckých tříd identické. Cílem kritéria sdílení linky je minimalizovat rozdíly mezi virtuálními časy sourozeneckých tříd. Při výpočtu časové náročnosti, termínu a virtuálního času využívá algoritmus tři křivky, pro každý parametr jednu: křivka časové náročnosti E_i , křivka termínu D_i a křivka virtuálního času V_i .

O algoritmu existuje několik pozoruhodných poukazů. Za prvé, jakmile HFSC potřebuje použít dvě kritéria pro výběr paketu pro podporu služeb sdílení linky a real-time, další hierarchický algoritmus HPFQ (Hierarchical Packet Fair Queueing) vybere paket pouze na základě kritérií sdílení linky a nyní je možné podporovat obě služby sdílení linky i real-time. To je z toho důvodu, že HPFQ zaručuje pouze služby lineární křivky a to je možné zaručit všem službám lineární křivky v celé hierarchii tříd. Pro porovnání, HFSC podporuje oddělené zpoždění a rozdělení šířky pásma garantováním služeb nelineárních křivek. Je ale nemožné zaručovat garanci služeb všech nelineárních křivek současně v celé hierarchii tříd. Z tohoto důvodu, HFSC používá dvě oddělená kritéria pro každý úkol sdílení linky a real-time, a využívá mechanismus časové náročnosti pro určení, které kritérium použít. Za druhé algoritmus používá tři typy časových parametrů: termín, časová náročnost, virtuální časy. Zatímco uzly list využívají všechny tři parametry, vnitřní uzly využívají pouze parametr virtuálního času. To proto, že termíny a časové náročnosti jsou používány za účelem zajištění servisních křivek a HFSC poskytuje servisní křivku pouze třídám typu list. Na druhou stranu jsou virtuální časy využívány pouze za účelem hierarchického sdílení linky, která zahrnuje celou hierarchii, a proto jsou tyto časy užívány ve všech třídách hierarchie. Třetím bodem povšimnutí je to, že všechny tři parametry jsou časové hodnoty, které jsou měřeny s pohledem na různé časové měřítko. Termíny a časové náročnosti jsou reálné v tom smyslu, že jsou měřeny s ohledem na fyzicky reálný čas. Absolutní hodnoty jsou důležité tak, jak potřebují být porovnány s reálným časem. Naproti tomu se virtuální čas měří s ohledem na celkovou hodnotu služeb poskytovaných rodičovskou třídou. Relativní odchylka virtuálních časů

sourozeneckých tříd je více důležitá, než jejich skutečná hodnota. Také je třeba vzít na vědomí, že kromě výhody oddělení zpoždění a rozdělení šířky pásma podporou služeb nelineárních křivek, HFSC poskytuje přesnější zpoždění pásma než HPFQ i pro hierarchie tříd pouze se službami lineárních křivek. Klíčovým poznatkem u HPFQ je, že plánování paketů je založeno výhradně na kritériu sdílení linky, které potřebuje směřovat rekurzivně z kořenové třídy ke třídě listu, kde vybírá další paket pro přenos. Čistým efektem je, že poskytované zpoždění třídy list vzrůstá s rostoucí hloubkou listové hierarchie. Pro porovnání u HFSC zpoždění třídy list klesá s vybraným kritériem real-time paket, který bere v úvahu pouze třídy typu list. Z tohoto důvodu je zpoždění nezávislé na hierarchii tříd [13].



OBRÁZEK 5.8 - HIERARCHICKÉ SDÍLENÍ LINKY

5.2.4.1 Časová náročnost a termín

V této části je popsáno jakým způsobem počítat termíny a časové náročnosti pro každou třídu typu list. Pro každou třídu i , algoritmus udržuje dvě křivky. Jednu pro každý parametr: křivka $E()$ je křivka časové náročnosti a křivka $D()$ je křivkou termínu. Kromě toho je zde obsažena proměnná C_i , která je zvyšována dle délky paketu v každém čase třídy i , paket je vybírán za použití kritéria real-time. Tímto způsobem C_i vyjadřuje celkovou hodnotu služeb, které byly přijaty třídou v čase vybírání paketů dle real-time kritérií. Křivka pro termín $D_i()$ je inicializována pro služby křivky $S_i()$, a je aktualizována v každé časové relaci i , která se stává aktivní v čase t_a dle následujícího vzorce:

$$D_i(t) = \min(D_i(t), S_i(t - t_a) + c_i), \forall t > D_i^{-1}(c_i).$$

ROVNICE 5.1 - ROVNICE KŘIVKY TERMÍNU $D_i()$

Vzhledem k tomu, že C_i nepočítá se službami přijatými relací skrz kritérium sdílení linky, termíny budoucích paketů nebudou ovlivněny i za předpokladu, že relace přijímá přebytečné služby

z hierarchie sdílení linky. Tato vlastnost je podstatou beztretnosti z hlediska spravedlivého sdílení. Jakmile jsou termíny použity pro garanci servisní křivky pro třídy typu list, časové náročnosti jsou použity pro rozhodování, které ze dvou plánovaných kritérií bude aplikováno pro výběr následujícího paketu. Klíčovým poznatkem je, že nelineární servisní křivka, není vždy schopna dosáhnout perfektního sdílení linky a garanci všech servisních křivek ve stejném čase. Typickým případem je, když servisní křivka s konkávním průběhem začínající v čase t_a je připojována k relaci s konvexní servisní křivkou. Před připojením relace i , přijmou další relace nadbytečné služby, ale jejich termínová křivka není aktualizována. Jakmile se aktivuje relace i , je součet hodnot termínových křivek všech aktivních relací vyšší než rychlost serveru. Je nemožné uspokojit servisní křivky všech relací. Jediným řešením je mít server přiděluje aktivním relacím službu „dost“ za pomoci kritéria real-time, takovým způsobem, že bude mít server dostatečnou kapacitu pro uspokojení všech servisních křivek do doby, než se aktivuje další relace. Nicméně je-li paket podán za použití kritéria real-time, ale následující paket má kratší virtuální čas, je odeslán z distribuce ideální sdílené linky. Z důvodu minimalizování rozdílu od modelu ideální sdílené linky je třeba podávat pakety pomocí používání kritéria sdílení linky, kde neexistuje nebezpečí, že garance pro třídy list budou v budoucnu porušeny [14].

U HFSC jsou časové náročnosti použity pro rozhodnutí, které ze dvou kritérií bude aplikováno pro výběr následujícího paketu. Pro lepší představu časové náročnosti: $E(t)$ je minimum služeb, které by měli být přijaty všemi relacemi v čase t , bez ohledu na příchozí datový tok, souhrnná doba služby vyžadována všemi relacemi v jakémkoliv budoucím časovém intervalu (t, t') nesmí přesáhnout $R \times (t' - t)$ tj. nesmí překročit kapacitu serveru R . Toto je nezbytná podmínka, pokud aktivní relace neobdrží alespoň službu $E(t)$ v čase t , poté následuje scénář, ve kterém bude servisní křivka alespoň jedné relace v budoucnosti narušena. Nejhorší scénář je, pokud jsou po dobu t aktivní všechny relace. Z důvodu, že výše uvedená podmínka platí i v budoucím čase t' existuje vzorec:

$$E(t) = \sum_{i \in A(t)} D_i(t) + [\max_{t' > t} (\sum_{i \in A(t')} (D_i(t') - D_i(t)) + \sum_{i \in P(t')} (D_i^*(t') - D_i^*(t)) - R \times (t' - t))]^+$$

ROVNICE 5.2 - KŘIVKA ČASOVÉ NÁROČNOSTI $E(t)$

Kde D_i^* reprezentuje křivku termínu pasivní relace i , která se stává aktivní v době t a $[x]^+$ označuje $\max(x, 0)$. Nejhorší případ je, když se všechny pasivní relace stanou aktivní v čase t , maximální požadovaná služba všemi relacemi během časového intervalu (t, t') , přičemž všechny z nich zůstávají aktivní:

$$\sum_{i \in A(t)} (D_i(t') - D_i(t)) + \sum_{i \in P(t)} (D_i^*(t') - D_i^*(t))$$

ROVNICE 5.3 - NEJHORŠÍ PŘÍPAD KDY SE VŠECHNY PASIVNÍ RELACE STANOU AKTIVNÍMI

Jelikož všechny relace mohou přijmout nejvíce $R \times (t' - t)$ služeb během intervalu (t, t') , a od doby t mohou aktivní relace přijmout nejméně : $\sum_{i \in A(t)} D_i(t)$ za účelem uspokojení servisní křivky.

$E(t)$ reprezentuje minimální služby, které by měly být přiděleny aktivním relacím v čase t využívající kritéria real-time s cílem zajistit garanci servisních křivek všech budoucích relací. Zbývající (přebytečné) služby mohou být přiděleny kritériím sdílení linky [14].

Algoritmus je optimální v tom smyslu, že může garantovat servisní křivky všech relací přidělováním přesné $E(t)$ služby v čase t . S ohledem na tuto možnost, by algoritmus jednoduše přidělil $E(t)$, a přerozdělil přebytečné služby v souladu s kritérii sdílení linky. Hlavním úkolem tohoto algoritmu je efektivní výpočet $E(t)$. Bohužel je to obtížné z několika důvodů: Prvním důvodem je, že $E(t)$ nezávisí pouze na křivkách termínu aktivních relací, ale také na křivkách termínů pasivních relací. Vzhledem k tomu, že termínová křivka závisí na čase, kdy se relace stane aktivní, je potřeba stále udržovat přehled nad všemi možnými změnami, které jsou v nejhorším případě úměrné počtu relací. Druhým důvodem je, že i když jsou všechny termínové křivky dvoudílné lineární, výsledná křivka $E(t)$ může být vícedílná lineární, což je obtížné na efektivní implementaci. Proto byla vybrána komplexnost za přesnost, nadhodnocením $E(t)$. První krok:

$$D_i^*(t') - D_i^*(t) \leq S_i(t' - t), \forall t' > t.$$

ROVNICE 5.4 - PRVNÍ KROK ODVOZENÍ KŘIVKY ČASOVÉ NÁROČNOSTI

Využitím této nerovnosti a faktu $\sum_i S_i(t) \leq R \times t$, pro všechny t začíná:

$$E(t) \leq \sum_{i \in A(t)} D_i(t) + [\max_{t' > t} (\sum_{i \in A(t)} (D_i(t') - D_i(t) - S_i(t' - t)))]^+ \leq \sum_{i \in A(t)} D_i(t) + [\max_{t' > t} (D_i(t') - D_i(t) - S_i(t' - t))]^+.$$

ROVNICE 5.5 - DRUHÝ KROK ODVOZENÍ KŘIVKY ČASOVÉ NÁROČNOSTI

Konečná definice křivky časové náročnosti je:

$$E_i(t) = D_i(t) + [\max_{t' > t} (D_i(t') - D_i(t) - S_i(t' - t))]^+, \\ \forall t > D^{-1}(c_i).$$

ROVNICE 5.6 - FINÁLNÍ DEFINICE KŘIVKY ČASOVÉ NÁROČNOSTI

Křivky časové náročnosti $E_i(t)$ určují maximální hodnotu služeb přijatých relací i v čase t pomocí kritérií real-time. Vzhledem k tomu, že $\sum_{i \in A(t)} E_i(t) \geq E(t)$ máme dostatečnou, ale nezbytnou podmínku.

$E_i(t)$ je aktualizována pokaždé, když se relace i stane aktivní pomocí funkce *update_EC* podle výše uvedeného vzorce. Je důležité poznamenat, že vzorec zabývající se algoritmy servisních křivek

libovolného tvaru vypadá komplikovaně, ale např. vzorec pro křivky časové náročnosti je vlastně docela jednoduché vypočítat. Například pro relaci s konkávní servisní křivkou je křivka časové náročnosti stejná jako křivka termínu. Minimální sazba služeb pro relace s konkávní servisní křivkou se nebudou v budoucnosti zvyšovat, což je způsobeno tím, že není nutné zajišťovat do budoucnosti služby těmito relacím. Podobně pro relace s dvoudílnou lineární servisní křivkou (první sklon α , druhý sklon β , kde $\alpha < \beta$), křivka časové náročnosti je lineární křivkou se sklonem β [14].

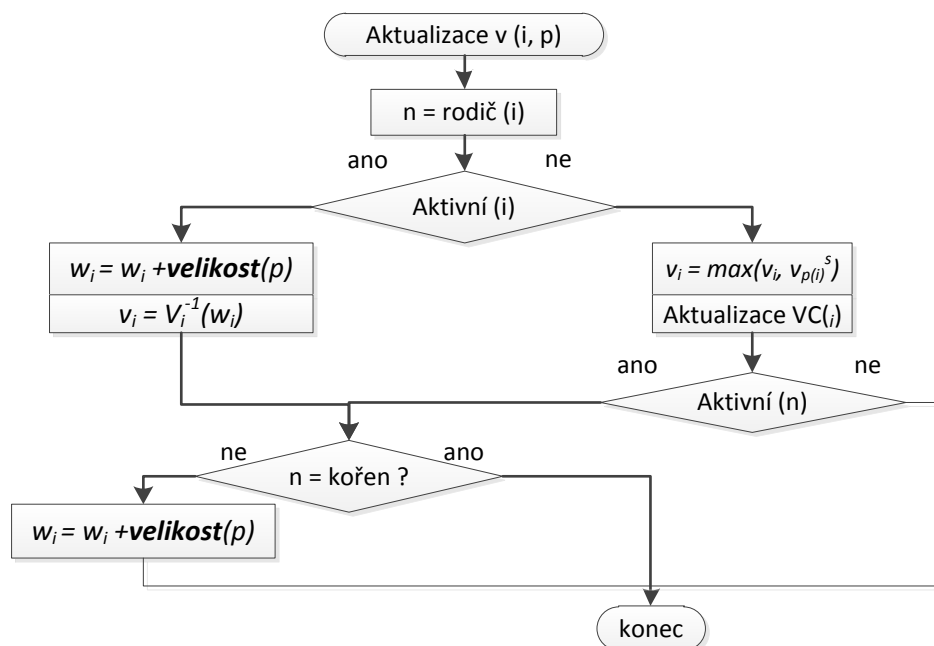
5.2.4.2 Virtuální čas

Koncept virtuálního času byl poprvé navržen v souvislosti s algoritmy spravedlivého řazení paketů (PFQ Packet Fair Queueing) a hierarchicky spravedlivého řazení paketů (HPFQ Hierarchical-PFQ) pro dosažení spravedlivosti, reálného času a hierarchického sdílení linky. V HFSC je používána zjednodušená verze virtuálního času pro dosažení hierarchického sdílení linky. Každý algoritmus spravedlivého řazení udržuje systém virtuálního času $v^i()$. Kromě toho udržuje v každé relaci i virtuální startovací čas $s_i()$ a virtuální finální čas $f_i()$. Intuitivně lze říct, že $v^i(t)$ reprezentuje normalizovanou hodnotu času služby, kterou by každá relace měla přijmout v čase t , $s_i(t)$ reprezentuje normalizovanou hodnotu času služby, kterou relace i přijala v čase t a $f_i(t)$ reprezentuje hodnotu mezi $v_i(t)$ a normalizovanou službou, kterou by měla relace i přijmout pro podávání paketů do čela její fronty. Vzhledem k tomu, že $s_i(t)$ sleduje služby přijaté relací i v čase t , tak $s_i(t)$ také vyvolává virtuální časy relace i , případně označené $v_i(t)$. Cílem algoritmů PFQ je minimalizovat rozdíly mezi $v_i(t)$ a $v(t)$. U HPFQ systému, každá třída udržuje funkci virtuálního času a cílem je minimalizovat rozdíly mezi všemi sourozeneckými uzly v hierarchii. Různé PFQ algoritmy jsou rozlišné ve dvou aspektech: systém počítá funkce virtuálního času a politika třídění paketů. Příklady systému funkce virtuálního času jsou počáteční časy právě podávaného paketu, finální časy právě podávaného paketu a nejkratší počáteční čas ze všech paketů načtených front. Příklady pravidel pro výběr paketů jsou: nejprve nejkratší počáteční čas (SSF Smallest Start time First), nejkratší finální čas (SFF Smallest Finish time First) a nejnižší časová náročnost (SEFF Smallest Eligible Finish time First). Volba různých systémů virtuálního času a pravidel výběru paketu bude mít vliv na reálný čas a vlastnosti spravedlivého třídění ve výsledném algoritmu PFQ.

H-FSC udržuje pro každou třídu v hierarchii funkci virtuálního času $v_i(t)$, která reprezentuje normalizovanou hodnotu časových služeb třídy i , které byly přijaty v čase t . U H-FSC je virtuální čas využíván kritérii pro sdílení linky k distribuci služeb podél hierarchií dle třídy servisní křivky. Kritéria sdílení linky jsou využívána pro výběr paketu pouze tehdy, nejsou-li použita kritéria real-time. Vzhledem k tomu, že garance služeb reálného času pro třídy typu list jsou zajišťovány výběrem

paketu pomocí kritérií real-time, je efekt na výkon tím, že jsou využity různé systémy funkcí virtuálního času a podmínky výběru paketu u kritérií sdílení linky, méně kritický. U HFSC je využíván systém výběru paketu SSF a následující systém funkce virtuálního času: $v_i^s = (v_{imin} + v_{imax}) / 2$, kde v_{imin} a v_{imax} jsou hodnoty maximálního a minimálního počátečního virtuálního času mezi všemi třídami i aktivovanými ve třídách potomků. Tím je zajištěno, že rozdíl mezi virtuálními časy dvou sourozenských relací je ohraničený. Je důležité poznamenat, že v_i^s může být buď v_{imin} nebo v_{imax} výsledkem v rozporu s počtem relací [14].

U H-FSC je $v_i(t)$ opakovaně počítáno pomocí předchozí funkce virtuálního času a servisní křivky relace. Virtuální časy jsou aktualizovány po dokončení služby paketu nebo poté co se třída stane aktivní. Vývojový diagram funkce pro aktualizaci virtuálního času je znázorněn zde:



OBRÁZEK 5.9 - VÝVOJOVÝ DIAGRAM AKTUALIZACE VIRTUÁLNÍHO ČASU

Funkce **aktualizace_v** rekurzivně aktualizuje virtuální čas a funkci virtuálního času na lince potomek-rodíč, dokud nedosáhne kořene nebo třídy rodiče, která byla aktivní v čase před časem t . V algoritmu je využívána funkce virtuální křivky $V_i()$ a inverzní funkce $v_i()$ namísto v_i . V_i je aktualizováno za použití funkce **aktualizace_VC** v každém čase, kdy se třída stává aktivní.

$$V_i(t) = \min(V_i(t)S_i(t - v_{p(i)}^s + w_i), \forall t > V_i^{-1}(w_i))$$

ROVNICE 5.7 - DEFINICE VIRTUÁLNÍ KŘIVKY

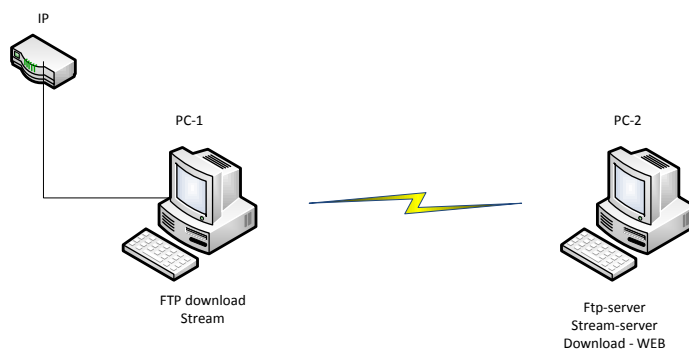
Kde w je konečná hodnota služeb přijatých v čase t a $v_{p(i)}^s$ je systémový virtuální čas pro třídu i typu rodič. Nakonec je třeba poznamenat, že pokud je $S_i()$ *přímka* se sklonem r_i , získá se rovnice ve tvaru $V_i(t) = r_i t$. Potom je virtuální čas v_i jednoduše zapsán $V_i^{-1}(w_i) = w_i / r_i$, což je přesný virtuální čas relace i v PFQ algoritmu [14].

6. QoS v OS Unixového typu

V této části bude probíhat testování podpory QoS v operačním systému Unixového typu distribuce UBUNTU 13.10 Saucy Salamander. Testování bude probíhat v několika navržených topologiích bezdrátových sítí. U všech zapojení bude při vytváření bezdrátové sítě použit standard 802.11 g.

6.1 Topologie se dvěma stanicemi, z nichž jedna slouží jako AP

První topologií pro ověření funkčnosti podpory QoS v operačních systémech Unixového typu, je zapojení zobrazené na obrázku 6.1. Obě stanice jsou vybaveny bezdrátovým USB adaptérem pro připojení a vytvoření bezdrátové sítě. Jedna ze stanic je propojena s venkovní IP sítí. Stanice PC-1 je připojena do sítě vedené v učebně SC5.32. Na obou stanicích je nainstalována v oracle vm virtualbou distribuce linuxu UBUNTU 13.10. Ke stanicím je připojen usb wireless adapter pro bezdrátovou komunikaci. Pro funkčnost adaptérů ve virtualboxu je třeba tyto adaptéry povolit. Horní lišta -> devices -> usb devices -> vybrat usb wireless adapter.



OBRAZEK 6.1- TOPOLOGIE SE DVĚMA STANICEMI, Z NICHŽ JEDNA SLOUŽÍ JAKO AP

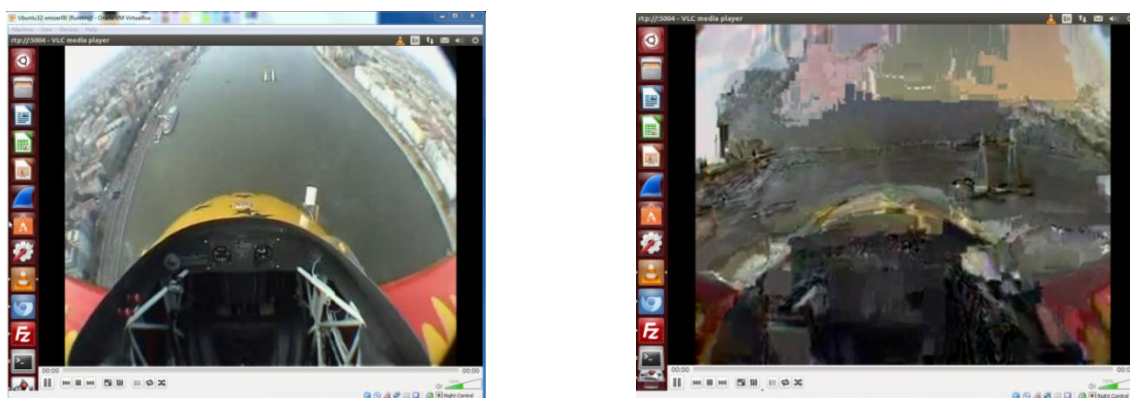
Pro datový provoz je využito, přenosu dat z PC-2, na kterém je nainstalován ftp server, na PC-1. Dále je využito streamování videa z PC-2 na PC-1 pomocí VLC media playeru, za použití grafického provedení programu IPERF nazývaného JPERF je vytvořena simulace VoIP hovoru na portu 5060, příkazem ping ověřujeme dobu odezvy mezi stanicemi a v poslední řadě probíhá download na stanici PC-2 z venkovní IP sítě.

Nastavení PC-1: Po spuštění je nejprve vytvořena bezdrátová síť. V nabídce připojení, která je v Ubuntu v pravém horním rohu, je zvoleno *create wireless network* -> zde se nastavuje název a zabezpečení nové sítě. Dále jsou spuštěny v PC-1 programy potřebné k vytvoření provozu a měření. Pro ftp přenos je použit program FileZilla, který je spojen s ftp serverem PC-2 pomocí ip adresy a přihlašovacích údajů. Otevření síťového streamu ve VLC media playeru se spustí v nabídce *media* -> *open network stream* -> *rtp://@: zvolený port*. Program jperf je třeba otevřít z terminálu a pro využití všech funkcí je nutné využít práv superuživatele. Pomocí příkazu *cd* se provede přepnutí do adresáře,

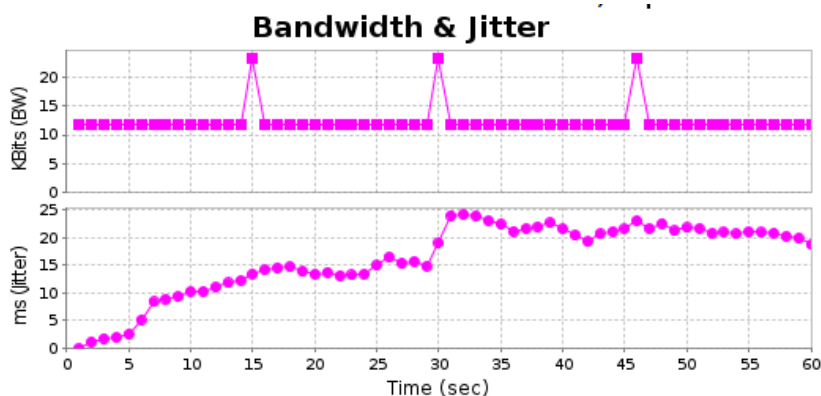
kde je jperf umístěn a dále pomocí příkazu `sudo sh jperf.sh` je program spuštěn. Ve výběru *mode*, je označena možnost *server* a nastaven port 5060, dalším nastavením je v části *transport layer option* označeno UDP -> `run iperf`.

Nastavení PC-2: Připojení k bezdrátové síti vytvořené stanicí PC-1. FTP server byl nainstalován a nastaven při prvním spuštění. Zprovoznění streamovaného videa v programu VLC: *MEDIA* -> *STREAM* -> *výběr videa* -> *stream* -> *next* -> *RTP / MPEG Transport Stream* -> *add* -> *<ip adresa PC-1>* -> *port* -> *stream*. Spuštění programu JPERF viz dříve. V programu jperf je nastaven, *mode client* -> *<server ip address>* -> *port*, v nabídce *transport layer option* UDP a *Bandwidth 12,5 kbytes/s*. Pro vytvoření toku dlouhého 60s je v nabídce *Application layer option* nastavena hodnota *Transmit* na 60s -> `run iperf`. V internetovém prohlížeči např. chromium je spuštěno stahování zvoleného souboru. Posledním nástrojem pro zatížení sítě je ping, který je spuštěn na 60s z terminálu pomocí příkazu `ping <ip adresa PC-1> -c 60`.

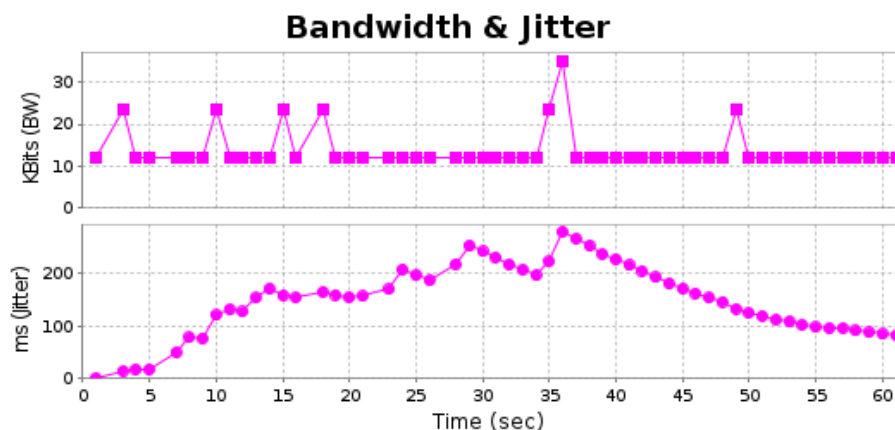
Na následujících obrázcích a v tabulce 6.1 jsou zobrazeny výsledky jednotlivých programů pro porovnání zatížené a nezatížené sítě bez nastavené jakékoliv podpory QoS.



OBRÁZEK 6.2- STREAM BEZ QOS VLEVO NEZATÍŽENÁ SÍŤ VPRAVO ZATÍŽENÁ SÍŤ



OBRÁZEK 6.3 - ŠÍŘKA PÁSMO A JITTERU U SIMULOVANÉHO VOIP BEZ ZATÍŽENÍ



OBRÁZEK 6.4 - ŠÍŘKA PÁSMO A JITTERU U SIMULOVANÉHO VOIP SE ZATÍŽENÍM

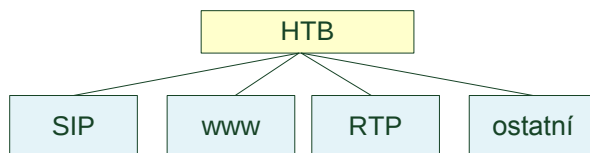
TABULKA 6.1 - HODNOTY NAMĚŘENÉ V ZAPOJENÍ 1 BEZ PODPORY QOS

zapojení bez podpory QoS		
	nezatížená	zatížená
ping	90 ms	702 ms
	0% ztráta	1% ztráta
ftp avg.	avg 500 KB/s	avg 240 KB/s
download	avg 6840 kbt/s	avg 1470 kb/s

Ze zobrazených hodnot lze usoudit, že při zatížení sítě je třeba přiřadit určitým datovým provozům prioritu, aby nedocházelo ke zpoždění a ztrátě některých paketů, které jsou např. u multimediálních služeb důležité.

Pro nastavení podpory QoS bude u první topologie využita metoda HTB a HFSC. Pro vytvoření podpory QoS je nutné vytvořit disciplínu qdisc a rozdělit datový provoz do jednotlivých tříd. Klasifikace provozu je prováděna pomocí filtrů a klasifikátoru u32.

Nastavení provozu na stanici PC-2 je zobrazeno níže. Jednotlivé příkazy se aplikují pomocí terminálu. Ve stanici je rozdělena šířka pásma do čtyř jednotlivých tříd viz obrázek 6.5.



OBRÁZEK 6.5 - ROZDĚLENÍ ŠÍŘKY PÁSMO HTB

Následující příkazy dělí šířku pásma 1000b/s do čtyř tříd. První třídě je přidělena šířka pásma 100b/s a je vyhrazena pro hovory SIP port 5060, což je pro hovory využívající VoIP dostačující. Druhé třídě je přiděleno pouze 100b/s, z toho důvodu, že třída je přidělena http přenosu a v testování je využíváno pouze downloadu. Pro třetí třídu je vyhrazeno 700b/s, a je přiřazena portu 5004 na kterém probíhá

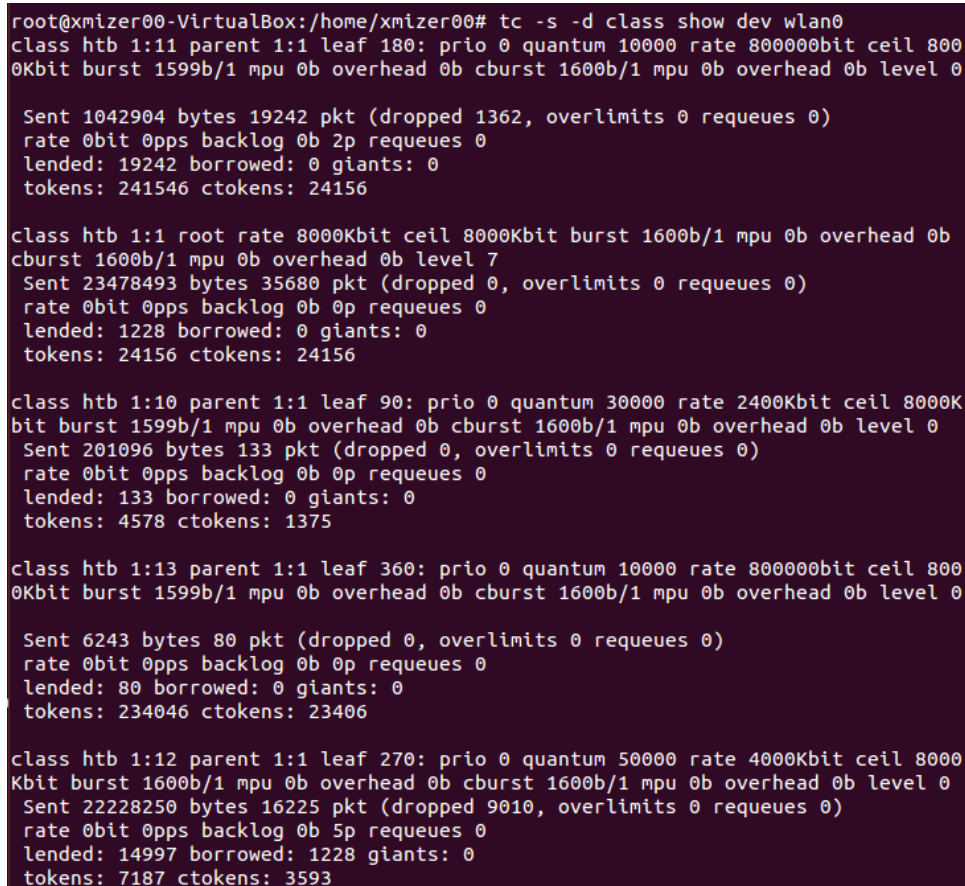
streamování videa. Zbýlá šířka pásma je přidělena ostatním službám. Graficky je rozdělení znázorněno na obrázku 6.5.

Postup vytvoření qdisc HTB a tříd typu list:

```
tc qdisc add dev wlan0 root handle 1: htb default 13
tc class add dev wlan0 parent 1: classid 1:1 htb rate 1000kbps ceil 1000kbps
tc class add dev wlan0 parent 1:1 classid 1:10 htb rate 100kbps ceil 1000kbps
tc class add dev wlan0 parent 1:1 classid 1:11 htb rate 100kbps ceil 1000kbps
tc class add dev wlan0 parent 1:1 classid 1:12 htb rate 700kbps ceil 1000kbps
tc class add dev wlan0 parent 1:1 classid 1:13 htb rate 100kbps ceil 1000kbps
tc filter add dev wlan0 protocol ip parent 1:0 prio 1 u32 match ip dport 5060 0xffff flowid 1:10
tc filter add dev wlan0 protocol ip parent 1:0 prio 1 u32 match ip sport 5060 0xffff flowid 1:10
tc filter add dev wlan0 protocol ip parent 1:0 prio 1 u32 match ip dport 80 0xffff flowid 1:11
tc filter add dev wlan0 protocol ip parent 1:0 prio 1 u32 match ip sport 80 0xffff flowid 1:11
tc filter add dev wlan0 protocol ip parent 1:0 prio 1 u32 match ip dport 5004 0xffff flowid 1:12
tc filter add dev wlan0 protocol ip parent 1:0 prio 1 u32 match ip sport 5004 0xffff flowid 1:12
```

v poslední řadě jsou přiřazeny metody qdisc pro jednotlivé třídy typu list.

```
tc qdisc add dev wlan0 parent 1:10 handle 30: sfq perturb 10
tc qdisc add dev wlan0 parent 1:11 handle 40: sfq perturb 10
tc qdisc add dev wlan0 parent 1:12 handle 50: sfq perturb 10
tc qdisc add dev wlan0 parent 1:13 handle 60: sfq perturb 10
```



```
root@xmizer00-VirtualBox:/home/xmizer00# tc -s -d class show dev wlan0
class htb 1:11 parent 1:1 leaf 180: prio 0 quantum 10000 rate 800000bit ceil 800
0Kbit burst 1599b/1 mpu 0b overhead 0b cburst 1600b/1 mpu 0b overhead 0b level 0

Sent 1042904 bytes 19242 pkt (dropped 1362, overlimits 0 requeues 0)
rate 0bit 0pps backlog 0b 2p requeues 0
lended: 19242 borrowed: 0 giants: 0
tokens: 241546 ctokens: 24156

class htb 1:1 root rate 8000Kbit ceil 8000Kbit burst 1600b/1 mpu 0b overhead 0b
cburst 1600b/1 mpu 0b overhead 0b level 7
Sent 23478493 bytes 35680 pkt (dropped 0, overlimits 0 requeues 0)
rate 0bit 0pps backlog 0b 0p requeues 0
lended: 1228 borrowed: 0 giants: 0
tokens: 24156 ctokens: 24156

class htb 1:10 parent 1:1 leaf 90: prio 0 quantum 30000 rate 2400Kbit ceil 8000K
bit burst 1599b/1 mpu 0b overhead 0b cburst 1600b/1 mpu 0b overhead 0b level 0
Sent 201096 bytes 133 pkt (dropped 0, overlimits 0 requeues 0)
rate 0bit 0pps backlog 0b 0p requeues 0
lended: 133 borrowed: 0 giants: 0
tokens: 4578 ctokens: 1375

class htb 1:13 parent 1:1 leaf 360: prio 0 quantum 10000 rate 800000bit ceil 800
0Kbit burst 1599b/1 mpu 0b overhead 0b cburst 1600b/1 mpu 0b overhead 0b level 0

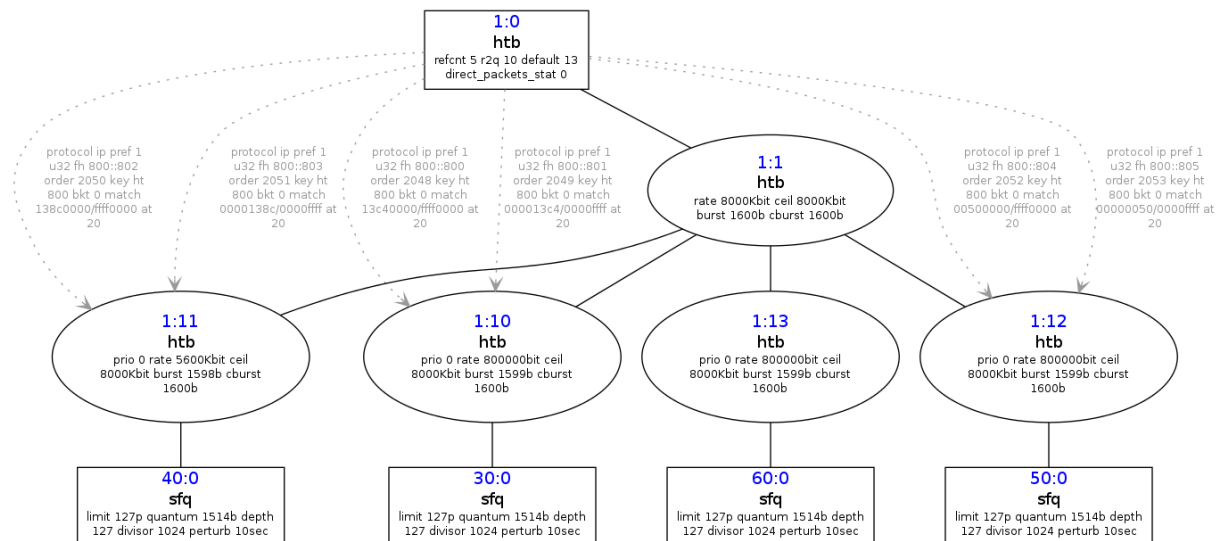
Sent 6243 bytes 80 pkt (dropped 0, overlimits 0 requeues 0)
rate 0bit 0pps backlog 0b 0p requeues 0
lended: 80 borrowed: 0 giants: 0
tokens: 234046 ctokens: 23406

class htb 1:12 parent 1:1 leaf 270: prio 0 quantum 50000 rate 4000Kbit ceil 8000
Kbit burst 1600b/1 mpu 0b overhead 0b cburst 1600b/1 mpu 0b overhead 0b level 0
Sent 22228250 bytes 16225 pkt (dropped 9010, overlimits 0 requeues 0)
rate 0bit 0pps backlog 0b 5p requeues 0
lended: 14997 borrowed: 1228 giants: 0
tokens: 7187 ctokens: 3593
```

OBRÁZEK 6.6 - VÝPIS NASTAVENÍ HTB

Na obrázku 6.6 je zobrazen výpis jednotlivých tříd, počet bytů odeslaných a přijatých těmito třídami a také výpis tokenů obsahující také počet zapůjčených a vypůjčených tokenů viz kapitola 5.2.3.

Grafické znázornění je možné vytvořit pomocí skriptu *tcviz.py*. Tento skript je volně šiřitelný a je třeba jej stáhnout. Pro získání grafického stromu pomocí skriptu je nutno následovat tento postup: *terminal -> cd <složka se skriptem> -> \$./tcviz.py dev | dot -Tpng >name.png*. Graf se vytvoří ve složce, ve které je umístěn skript *tcviz.py*. Na obrázku 6.7 je zobrazen strom vytvořený metodou HTB.

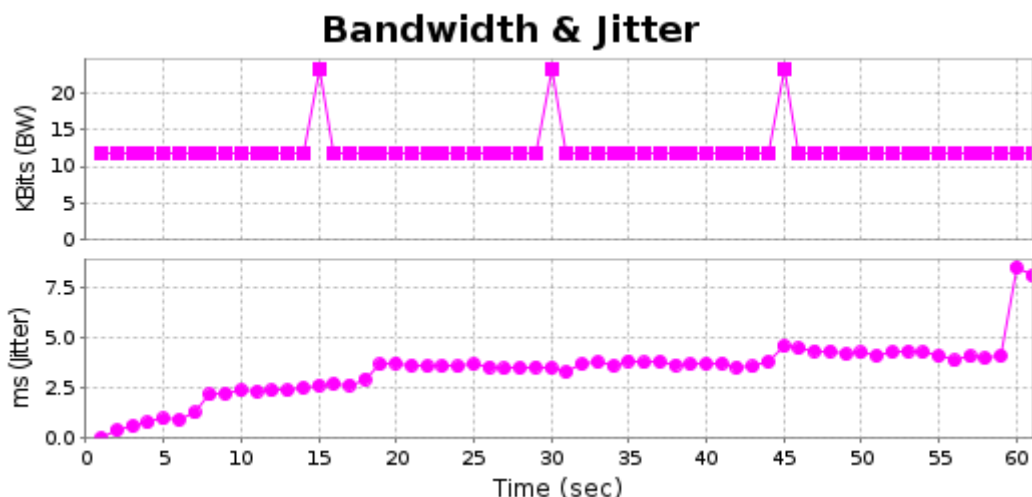


OBRÁZEK 6.7 - ROZDĚLENÍ HTB

V následujících obrázcích je zobrazen výsledek po rozdělení datového provozu do tříd a přiřazení filtrů k těmto třídám. Postup měření probíhal stejně jako v zapojení bez nastavených parametrů podpory QoS.



OBRÁZEK 6.8 - STREAM VIDEO ZE ZAPOJENÍ S NASTAVENÝMI PRIORITAMI HTB



OBRÁZEK 6.9 - ŠÍŘKA PÁSMO A JITTER S NASTAVNOU PODPOROU QOS POMOCÍ HTB

Z výše uvedených obrázků lze pozorovat rozdíly v přenosu při zatížené síti datovým tokem s použitím metody HTB, která rozděluje tok do tříd a přiřazuje jim jednotlivé filtry pro klasifikaci provozu. Z porovnání grafů v obrázcích 6.4 a 6.9, lze pozorovat rozdíly mezi šířkami pásma a jitterem bez implementované metody HTB a s využitím této metody. Na křivce jitteru je viditelný výrazný rozdíl, což je požadavkem využívání těchto metod při datovém provozu.

Další metodou pro srovnání je metoda HFSC. Postup měření je shodný s využitím stejných softwarových doplňků. U metody HFSC je možnost nastavení maximálního zpoždění, které je nastaveno v dělení provozu do tříd. Rozdělení datového provozu a přiřazení filtrů dle obrázku 6.5 za využití terminálu je zobrazeno v následující části:

```
tc qdisc add dev wlan0 root handle 1: hfsc default 13
tc class add dev wlan0 parent 1: classid 1:1 hfsc sc rate 1000kbit ul rate 1000kbit
tc class add dev wlan0 parent 1:1 classid 1:10 hfsc sc umax 1500b dmax 30ms rate 100kbit ul rate 1000kbit
tc class add dev wlan0 parent 1:1 classid 1:11 hfsc sc umax 1500b dmax 200ms rate 700kbit ul rate 1000kbit
tc class add dev wlan0 parent 1:1 classid 1:12 hfsc sc rate 100kbit ul rate 1000kbit
tc class add dev wlan0 parent 1:1 classid 1:13 hfsc sc rate 100kbit ul rate 1000kbit
tc filter add dev wlan0 protocol ip parent 1:0 prio 1 u32 match ip sport 5060 0xffff flowid 1:10
tc filter add dev wlan0 protocol ip parent 1:0 prio 1 u32 match ip dport 5060 0xffff flowid 1:10
tc filter add dev wlan0 protocol ip parent 1:0 prio 2 u32 match ip sport 5004 0xffff flowid 1:11
tc filter add dev wlan0 protocol ip parent 1:0 prio 2 u32 match ip dport 5004 0xffff flowid 1:11
tc filter add dev wlan0 protocol ip parent 1:0 prio 3 u32 match ip sport 80 0xffff flowid 1:12
```

```
tc filter add dev wlan0 protocol ip parent 1:0 prio 3 u32 match ip dport 80 0xffff flowid 1:12
```

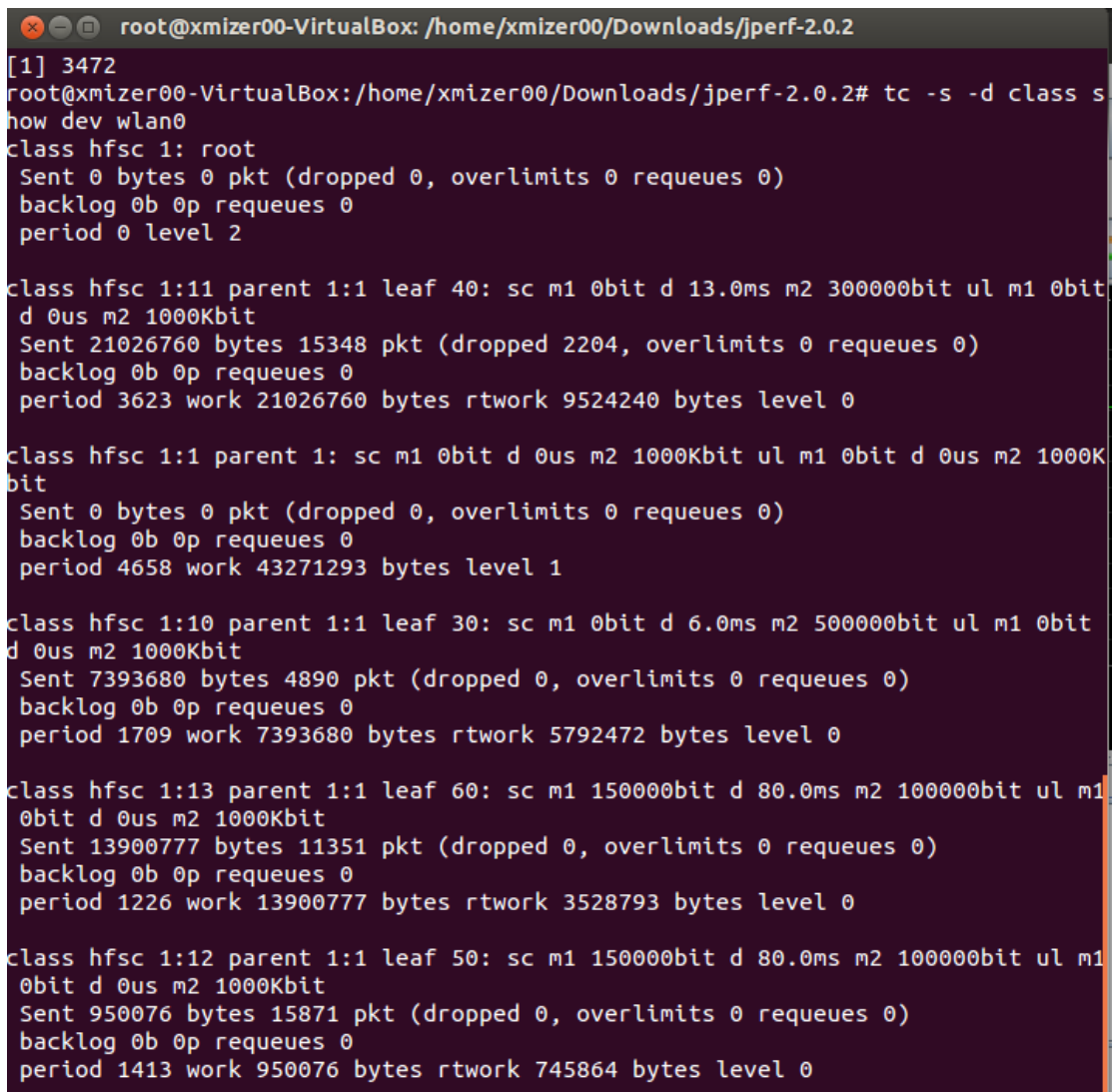
```
tc qdisc add dev wlan0 parent 1:10 handle 30: sfq perturb 10
```

```
tc qdisc add dev wlan0 parent 1:11 handle 40: sfq perturb 10
```

```
tc qdisc add dev wlan0 parent 1:12 handle 50: sfq perturb 10
```

```
tc qdisc add dev wlan0 parent 1:13 handle 60: sfq perturb 10
```

Výpis nastavení HFSC v terminálu je proveden příkazem `tc -s -d class show dev wlan0` a je zobrazen na obrázku 6.10. Je zde možno zpozorovat kolik dat prošlo jednotlivými třídami.



```
root@xmizer00-VirtualBox: /home/xmizer00/Downloads/jperf-2.0.2
[1] 3472
root@xmizer00-VirtualBox:/home/xmizer00/Downloads/jperf-2.0.2# tc -s -d class show dev wlan0
class hfsc 1: root
  Sent 0 bytes 0 pkt (dropped 0, overlimits 0 requeues 0)
  backlog 0b 0p requeues 0
  period 0 level 2

class hfsc 1:11 parent 1:1 leaf 40: sc m1 0bit d 13.0ms m2 300000bit ul m1 0bit d 0us m2 1000Kbit
  Sent 21026760 bytes 15348 pkt (dropped 2204, overlimits 0 requeues 0)
  backlog 0b 0p requeues 0
  period 3623 work 21026760 bytes rtwor 9524240 bytes level 0

class hfsc 1:1 parent 1: sc m1 0bit d 0us m2 1000Kbit ul m1 0bit d 0us m2 1000Kbit
  Sent 0 bytes 0 pkt (dropped 0, overlimits 0 requeues 0)
  backlog 0b 0p requeues 0
  period 4658 work 43271293 bytes level 1

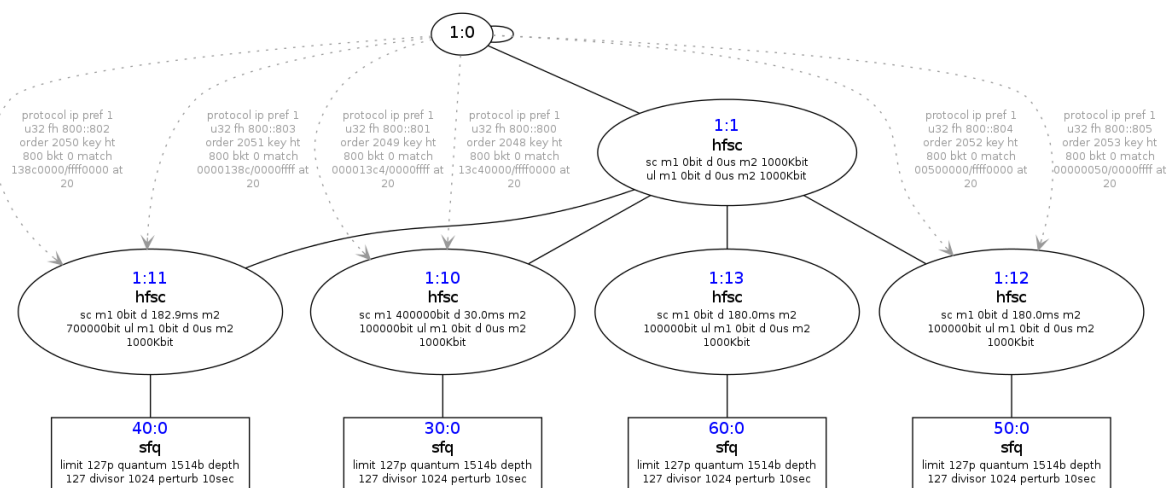
class hfsc 1:10 parent 1:1 leaf 30: sc m1 0bit d 6.0ms m2 500000bit ul m1 0bit d 0us m2 1000Kbit
  Sent 7393680 bytes 4890 pkt (dropped 0, overlimits 0 requeues 0)
  backlog 0b 0p requeues 0
  period 1709 work 7393680 bytes rtwor 5792472 bytes level 0

class hfsc 1:13 parent 1:1 leaf 60: sc m1 150000bit d 80.0ms m2 100000bit ul m1 0bit d 0us m2 1000Kbit
  Sent 13900777 bytes 11351 pkt (dropped 0, overlimits 0 requeues 0)
  backlog 0b 0p requeues 0
  period 1226 work 13900777 bytes rtwor 3528793 bytes level 0

class hfsc 1:12 parent 1:1 leaf 50: sc m1 150000bit d 80.0ms m2 100000bit ul m1 0bit d 0us m2 1000Kbit
  Sent 950076 bytes 15871 pkt (dropped 0, overlimits 0 requeues 0)
  backlog 0b 0p requeues 0
  period 1413 work 950076 bytes rtwor 745864 bytes level 0
```

OBRÁZEK 6.10 - VÝPIS METODY HFSC

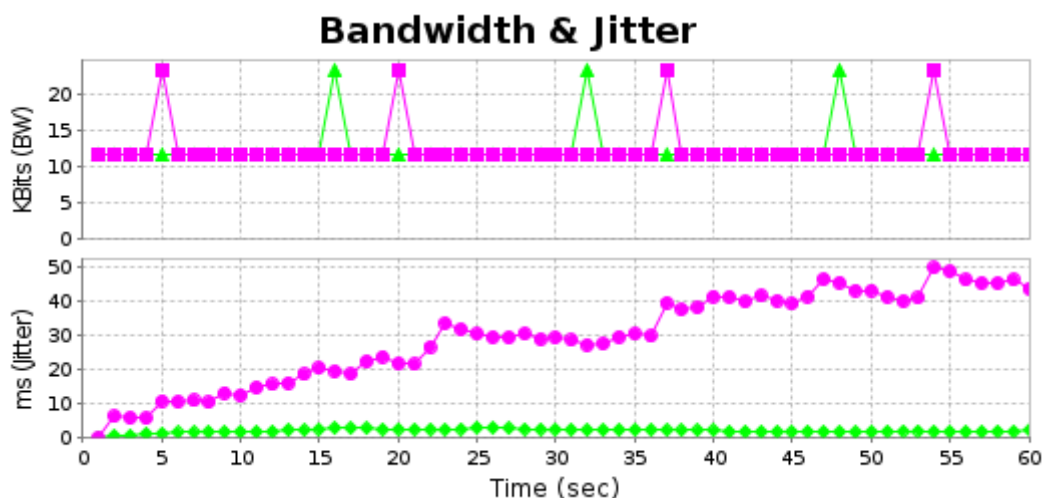
Rozdělení datového toku pomocí metody HFSC do grafického znázornění bylo provedeno stejně jako v předchozím případě. Graf Stromové struktury je zobrazen na obrázku 6.11.



OBRÁZEK 6.11 - ROZDĚLENÍ HFSC

Ve schématu je zobrazeno rozdělení do jednotlivých tříd s maximálními hodnotami šířky pásma a zpoždění, filtry a koncové metody qdisc v samotných třídách.

Naměřené výsledky šířky pásma a jitteru jsou zobrazeny v obrázku 6.12. Fialová křivka v tomto grafu zobrazuje průběh bez nastavení jakékoliv podpory, zatímco zelená křivka zobrazuje průběh s využitím metody HFSC. Snímek z přenášeného streamu videa lze použít obrázek 6.8, protože video bylo plynulé a reálně pozorovatelné.



OBRÁZEK 6.12- ŠÍŘKA PÁSM A JITTER S HFSC(ZELENÁ) A BEZ HFSC (FIALOVÁ)

TABULKA 6.2 – DOBY ODEZVY A RYCHLOSTI FTP PŘENOSU U TOPOLOGIE 1

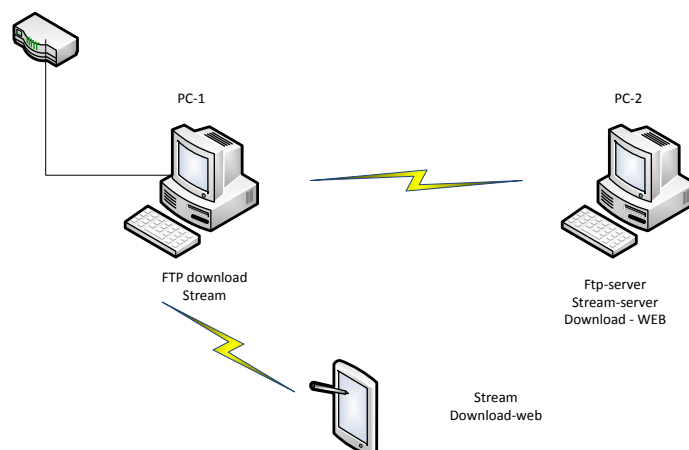
	bez QoS	s QoS
odezva mezi PC-1 a PC-2	211ms	62ms
průměrná rychlost ftp	380kB/s	105kB/s

V tabulce 6.2 jsou zobrazeny výsledky změny doby odezvy a průměrné rychlosti ftp přenosu u topologie jedna v porovnání s využitím metody HFSC a bez této metody. Z výsledku je zřejmé, že se

rychlost ftp přenosu snížila a to z toho důvodu, aby byla uvolněna přenosová šířka pásma pro multimediální přenosy, které byly metodou HFSC upřednostněny.

6.2 Topologie se třemi stanicemi, z nichž jedna slouží jako AP

Druhá zvolená topologie je podobného typu jako v předchozím případě viz obrázek 28. Do sítě je navíc připojeno třetí zařízení (zde smartphone SONY XPERIA S), na kterém probíhá stahování dat z internetu a také je na něm přehráváno streamované video vysílané ze stanice PC-2.

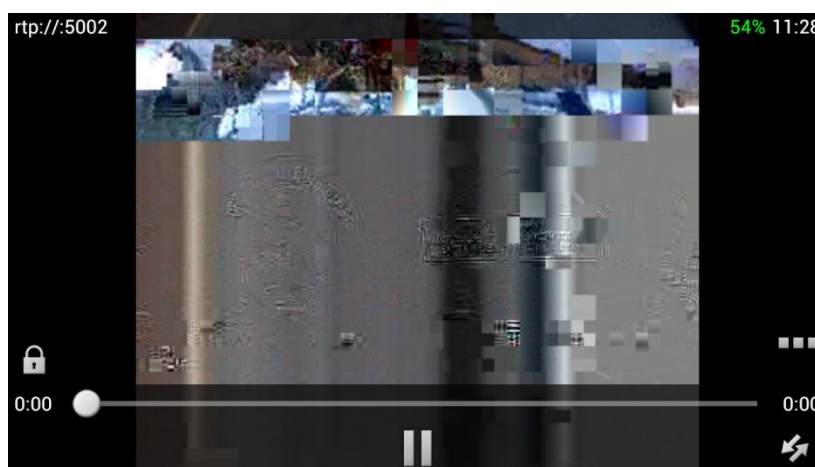


OBRÁZEK 6.13 - TOPOLOGIE SE TŘEMI STANICEMI, Z NICHŽ JEDNA SLOUŽÍ JAKO AP

Připojení mobilního telefonu do sítě nebylo možné ihned. Pro připojení mobilního zařízení je potřeba nainstalovat software *hostapd (Wireless Authenticator and Authentication Server)* ve stanici PC-1. Instalace: `terminal -> sudo apt-get install hostapd`. Nejprve je třeba nastavit konfigurační soubor programu: `terminal -> cd /etc/hostapd/ -> nano (vi, gedit, ...) hostapd.conf -> nastavení dle sítě vytvořené stanicí PC-1 (SSID, interface, ...) -> ctrl+x (uložení)`. Po nastavení konfiguračního souboru, je třeba mít ve stanici PC-1 vytvořenu síť se shodným názvem a případně i shodným heslem. Dalším krokem pro zpřístupnění sítě pro mobilní telefon je spuštění programu *hostapd*, které se provede následujícím způsobem: `terminál -> cd /etc/hostapd/ -> sudo hostapd hostapd.conf`. Nyní je možné se připojit s mobilním telefonem k bezdrátové síti vytvořené stanicí PC-1

Provoz mezi stanicemi PC-1 a PC-2 byl vytvořen stejně jako v předchozím případě. Pro streamování videa je třeba přidat ve stanici PC-2 další zařízení, které je připojené do sítě. Toto je provedeno v nastavení streamu v programu VLC viz kap. 6.1, pouze není zvolena metoda add jednou ale po nastavení první stanice, se přidá další zařízení pomocí `add -> <IP adresa zařízení> -> port`. Zvolený port bude jiný než pro zařízení PC-1 např: 5002 Pro přehrání streamu v mobilním zařízení je postup stejný jako v kap. 6.1. Dále bylo v mobilním zařízení spuštěno stahování z internetu.

V této topologii, ale i v dalších, bude využívána metoda HFSC, a to z toho důvodu, že metoda HFSC podporuje nastavení maximálního zpoždění, které je u přenosu multimediálních služeb vyžadováno. Ve stanici PC-2 byl datový provoz rozdělen do stejných skupin a přiřazeny stejné filtry jako v předchozí topologii. Na obrázku 6.14 je zobrazen výsledek ze streamovaného videa přehrávaného v mobilním telefonu v případě že, je nastavení HFSC shodné s předchozí kapitolou 6.1. Toto nastavení nemá pro podporu streamovaného videa nastaven port 5002, na kterém je zapnuto streamování pro mobilní stanici, ale pouze pro port 5004, který je využíván na stanici 5004.



OBRÁZEK 6.14 - STREAM VIDEO NA MOBILNÍM TELEFONU BEZ PODPORY QOS

Video přenášené streamem bylo na mobilním telefonu nesledovatelné, bylo kostkované, často se pozastavovalo a zvuk byl nerozpoznatelný a trhaný. Pro vyzkoušení podpory QoS v této topologii byl přenastaven ve stanici PC-2 filtr pro datový tok s ID 1:11 na port 5002, na kterém probíhá stream videa pro mobilní telefon. Stejný filtr a qdisc musela být přiřazena i na stanici PC-1, protože mobilní stanice byla připojena k bezdrátové síti tvořené touto stanicí. V této stanici bylo nastaveno upřednostnění pouze pro port 5002, a zbytek šířky pásma je pro ostatní datové toky.

Nastavení PC-1:

```
tc qdisc add dev wlan0 root handle 1: hfsc default 11
```

```
tc class add dev wlan0 parent 1: classid 1:1 hfsc sc rate 1000kbit ul rate 1000kbit
```

```
tc class add dev wlan0 parent 1:1 classid 1:10 hfsc sc umax 1500b dmax 200ms rate 700kbit ul rate 1000kbit
```

```
tc class add dev wlan0 parent 1:1 classid 1:11 hfsc sc rate 300kbit ul rate 1000kbit
```

```
tc filter add dev wlan0 protocol ip parent 1:0 prio 2 u32 match ip sport 5002 0xffff flowid 1:10
```

```
tc filter add dev wlan0 protocol ip parent 1:0 prio 2 u32 match ip dport 5002 0xffff flowid 1:10
tc qdisc add dev wlan0 parent 1:10 handle 20: sfq perturb 10
tc qdisc add dev wlan0 parent 1:11 handle 30: sfq perturb 10
```

Na stanici PC-2 došlo ke změně pouze u filtru 1:11:

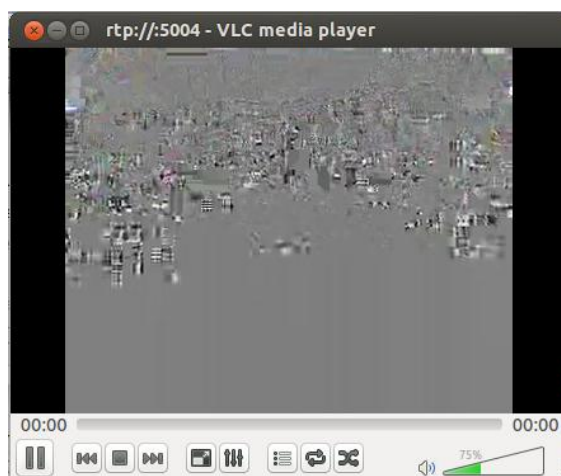
```
tc filter add dev wlan0 protocol ip parent 1:0 prio 2 u32 match ip sport 5002 0xffff flowid 1:11
tc filter add dev wlan0 protocol ip parent 1:0 prio 2 u32 match ip dport 5002 0xffff flowid 1:11
```



OBRÁZEK 6.15 - STREAM VIDEA NA MOBILNÍ TELEFON S OPDPOROU QOS

Obrázek 6.15 zobrazuje velký rozdíl mezi přenášeným videem s nastavenou podporou QoS a bez této podpory viz obrázek 6.14. Video i zvuk po nastavení podpory QoS, je plynulý a obraz je v kvalitě, která je k vidění v originálním videu.

Změny v nastavení podpory QoS ve stanici PC-2 způsobily, odebrání priorit pro stream videa na stanici PC-1, což je zobrazeno na obrázku 6.16.



OBRÁZEK 6.16 - STREAM VIDEA NA STANICI PC-1

Port 5004 v tomto nastavení spadá do skupiny datového toku pro zbylé přenosy, a tím pádem není pro přenos videa podporována dostatečná šířka pásma a maximální doba zpoždění.

Pomocí tohoto zapojení bylo ověřeno jaký vliv má nastavení šířky pásma pro různé porty, v síti s více stanicemi. Toto bylo dokázáno za využití streamování videa a přiřazování priorit pro porty na různá zařízení. Tabulka 6.3 obsahuje hodnoty doby odezvy a rychlosti ftp přenosu probíhajícího mezi stanicemi PC-1 a PC-2. V porovnání s tabulkou 6.2 je zřejmé, jak třetí připojené zařízení ovlivnilo tyto datové toky.

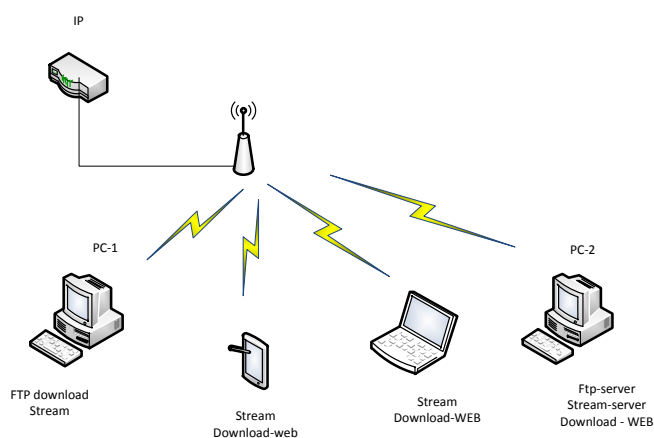
TABULKA 6.3 - DOBY ODEZVY A RYCHLOSTI FTP PŘENOSU U TOPOLOGIE 2

	bez QoS	s QoS
odezva mezi PC-1 a PC-2	1338ms	112ms
průměrná rychlost ftp	120kB/s	65kB/s

U hodnot s použitím metody HFSC je pozorovatelná změna doby odezvy téměř na dvojnásobek a u rychlosti ftp přenosu se rychlost téměř o polovinu snížila s porovnáním s předchozím zapojením. Je to způsobeno přiřazením priority multimediálním přenosům a také zvýšením datového toku, které je provedeno vysíláním streamu do mobilního telefonu. Hodnoty bez jakéhokoliv využití podpory QoS se hodnota doby odezvy několikanásobně zvýšila, což bylo způsobeno vyšším datovým tokem mezi stanicemi PC-1 a PC-2. Který byl navýšen streamem videa pro mobilní zařízení. Stejná příčina byla důvodem snížení rychlosti ftp přenosu, která byla v předchozím zapojení téměř třikrát vyšší než v zapojení z obrázku 6.13

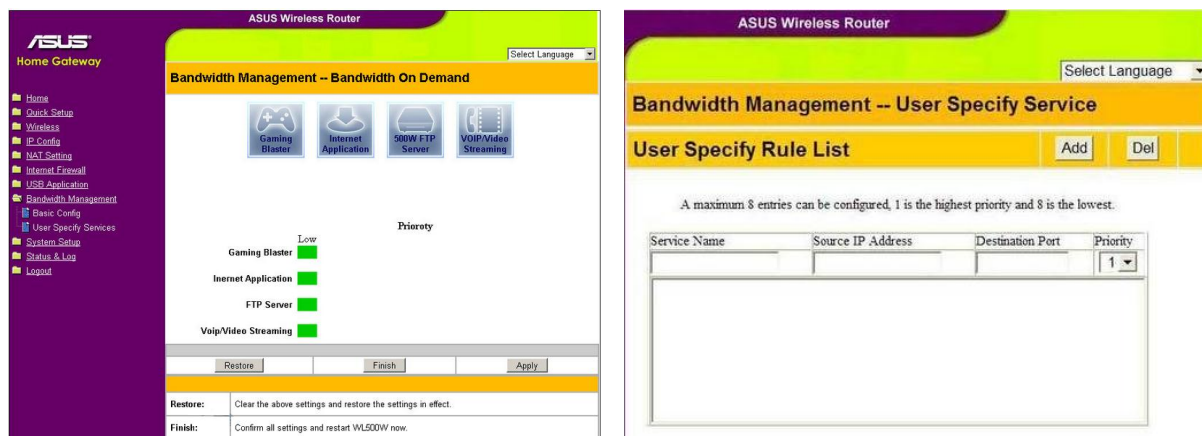
6.3 Topologie vytvoření sítě pomocí Access Pointu

Následující zapojení již nevyužívalo jako přístupový bod počítačovou stanici, ale byl využit přístupový bod ASUS WL-500W pro vytvoření bezdrátové sítě.



OBRÁZEK 6.17 - TOPOLOGIE S PŘÍSTUPOVÝM BODEM

Jako AP je v této topologii použit běžný přístupový bod, který nemá rozsáhlé možnosti rozdělení šířek pásma a přiřazení priorit. Toto zařízení dokáže pouze rozdělit datový tok a upřednostnit jej ve frontě k odeslání. Možnosti nastavení rozdělení šířky pásma viz obrázek 6.18.



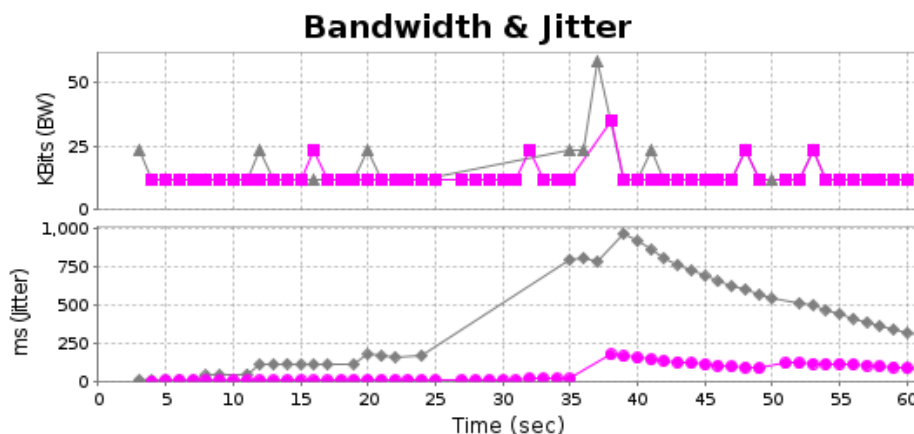
OBRAZEK 6.18 - MOŽNOSTI ROZDĚLENÍ ŠÍŘKY PÁSMAS ASUS WL-500W

Tento přístupový bod umožňuje buď grafickou možnost rozdělení šířky pásma (na obrázku 6.18 vlevo) nebo specifitější rozdělení, které umožňuje přidělovat priority pomocí zdrojové adresy a portu (na obrázku 6.18 vpravo). Ani toto nastavení ovšem není plnohodnotné a neumožňuje vytvořit rozdělení šířky pásma podle požadavků.

Ve zvolené topologii bude otestována součinnost nastavení podpory u koncových uživatelů bez jakéhokoliv nastavení podpory QoS v AP, ale pouze v koncových stanicích. Dále nastavení pouze v AP a v posledním případě nastavení podpory QoS u AP i u koncových uživatelů.

Pro vytvoření sítě je nejprve připojen AP ke stanici PC-1 a pomocí webového rozhraní je nastaven tak, aby tvořil bezdrátovou síť typu g a fungoval jako AP. Připojení k zařízení se provede pomocí prohlížeče a adresy 192.168.1.1, přihlašovací údaje *jméno*: admin *heslo*: admin. V nabídce *wireless* je nastaven název sítě *SSID*, kanál *auto*, typ sítě *only g* a také je zde možno nastavit zabezpečení. Po vytvoření jsou zařízení připojena k síti vytvořené přístupovým bodem a je mezi nimi vytvořen datový provoz jako v předchozích případech. Spojení stanic PC-1 a PC-2 pro ftp přenos probíhá stejně, pouze se změněním adresy serverové stanice. Stejná změna proběhne u streamování videa ze stanice PC-2, která bude navíc provozovat stream i do další přidané stanice, kterou zde zastupuje notebook Toshiba. Notebook navíc stahuje data z internetu.

Na grafech v obrázku 6.19 je vidět rozdíl průběhu šířky pásma a jitteru mezi stanicemi PC-1 a PC-2, kde šedá křivka zobrazuje průběh bez vytvořených tříd *tc* ve stanici PC-2 a růžová křivka představuje průběh po vložení shodného skriptu pro metodu HFSC jako v předchozích topologiích do terminálu.



OBRÁZEK 6.19 – SIMULOVANÝ VOIP S METODOU HFSC VE STANICI PC-2(FIALOVÁ) A BEZ METODY HFSC(ŠEDÁ)

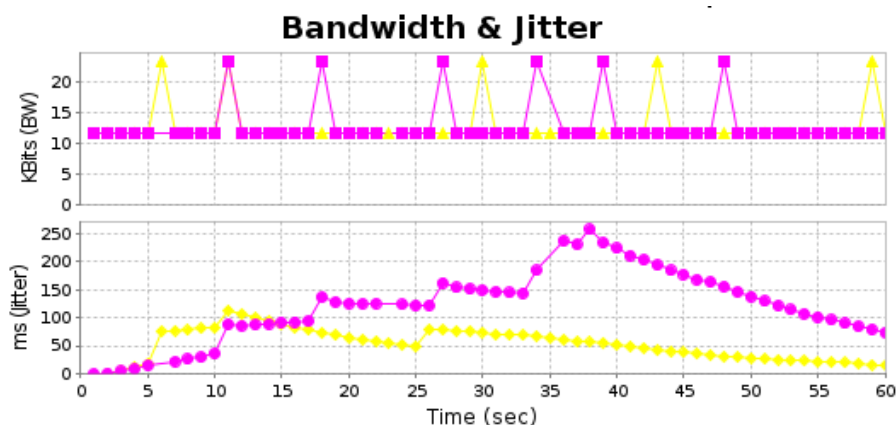
Video ve stanici PC-1 probíhá vcelku plynule a někdy je mírně rozmazané, ale je stále bez větších problémů pozorovatelné. V mobilní stanici a v notebooku je video absolutně nesledovatelné, je téměř celou dobu zastaveno a někdy se mírně posune. Toto je způsobeno tím, že ve stanici PC-2 je v nastavení přidělena šířka pásma 700 kb/s pro port 5004, na kterém je vyslán stream videa do stanice PC-1. Grafy na obrázku 6.19 znázorňují velký rozdíl mezi přenosem s nastavenými hodnotami metody HFSC a bez jakéhokoliv nastavení, to ale stále není ideální a to z toho důvodu, že AP nepodporuje žádnou klasifikaci datového toku a přidělování priorit. Na křivce jitteru je vidět, že při podpoře QoS metodou HFSC ve vysílající stanici, se hodnoty stále pohybují až ke 200ms. Zlepšení je vytvořeno pomocí klasifikace provozu a nastavení hodnot pro uplink ve vysílající stanici, což je provedeno na síťové vrstvě, tak přístupový bod pracující na vrstvě linkové tento datový tok opět nijak neklasifikuje a odesílá jej metodou best effort, to je důvodem zvýšených hodnot jitteru a občasného pozastavení a rozmazání videa ve stanici PC-1. U dalších dvou stanic je video pozastaveno, protože tomu zabraňuje již nastavení na straně serveru tedy ve stanici PC-2.

Dalším krokem je ověření funkčnosti podpory kvalitativních požadavků služeb při povolení metody WMM v přístupovém bodu bez nastavení metody HFSC a s nastavením metody HFSC ve stanici PC-2. Pro povolení metody WMM v přístupovém bodu je třeba se připojit k webovému rozhraní adresou 192.168.1.1, následně ve složce *wireless* se přepnout do podsložky *advanced* a zde přepnout položku *Enable WMM* na *Enabled* viz obrázek 6.20. Následuje -> *finish* -> *save & restart*.



OBRÁZEK 6.20 - POVOLENÍ METODY WMM

Po připojení všech zařízení k síti tvořené přístupovým bodem je nejprve odstraněna metoda HFSC ze stanice PC-2 pomocí příkazu: `tc qdisc del dev wlan0 root handle 1: hfsc default 13`. Pro ověření funkčnosti WMM je nutno vytvořit datový provoz jako v předchozích případech. Výsledné křivky simulovaného VoIP hovoru jsou na obrázku 6.21.

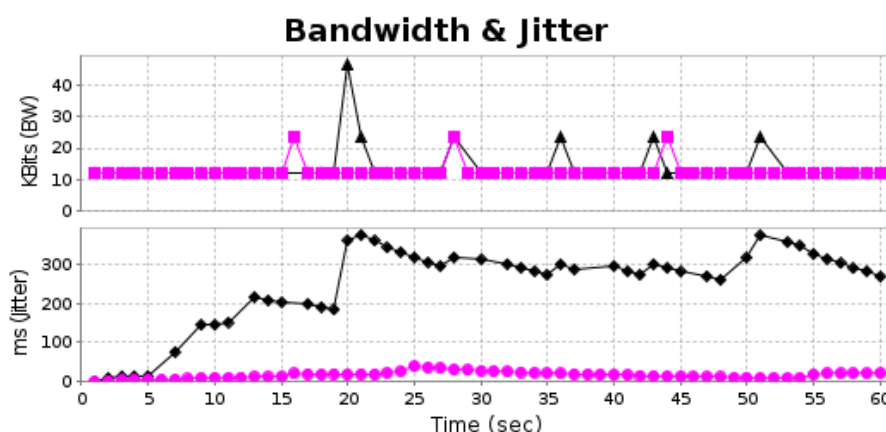


OBRÁZEK 6.21 - SIMULOVANÝ VOIP S POVOLENOU SLUŽBOU WMM S(ŽLUTÁ) A BEZ METODY HFSC(FIALOVÁ)

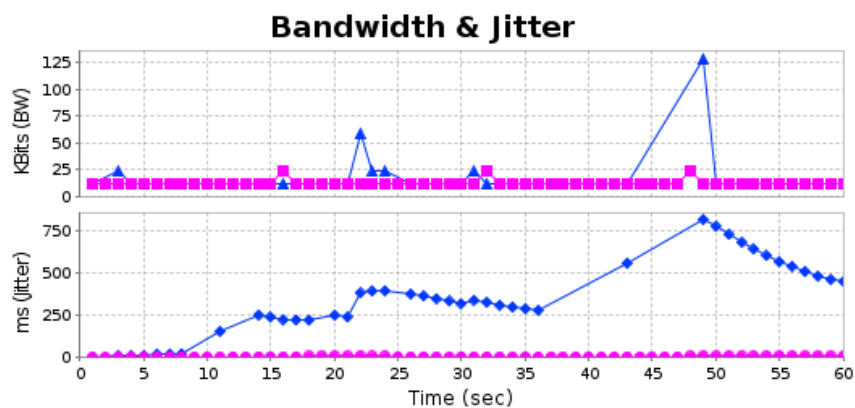
V grafech na obrázku 6.21 jsou zobrazeny křivky šířky pásma a jitteru pro zapojení obrázek 6.17, kde fialové křivky jsou s povolenou službou WMM v přístupovém bodu bez nastavené metody HFSC a žluté křivky s povolenou službou WMM v přístupovém bodu, ale zároveň s nastavenou metodou HFSC ve stanici PC-2. Z výsledků lze zjistit, že v prvním případě pouze s nastavenou službou WMM jsou výsledky podobné předchozímu grafu viz obrázek 6.19, kde byla použita pouze metoda HFSC bez povolené služby WMM. Hodnoty jitteru jsou u obou křivek v okolí hodnoty 200ms. Ve druhém případě při povolení WMM a zároveň nastavení metody HFSC ve stanici PC-2, je viditelné, že maximální hodnota jitteru klesla asi na polovinu. To je způsobeno řízením datového toku z vysílající stanice PC-2 pomocí nastavené metody HFSC a zároveň podporou metody WMM na další vrstvě, která dělí datový tok v přístupovém bodu a přiděluje prioritu požadovaným přenosům. Přenášené streamy videa byly v případě pouze se službou WMM ve všech stanicích někdy mírně rozmazané s občasným pozastavením, ve druhém případě bylo video plynulé ve stanici PC-1 a v dalších stanicích bylo nesledovatelné z důvodu nastavení hodnoty uploadu metodou HFSC ve stanici PC-2. Rychlost FTP přenosu se zapnutím metody HFSC snížila z původních cca 70kB/s na hodnoty pohybující se mezi 5–10 kB/s.

Hodnota jitteru pohybující se kolem 100ms není stále pro přenos VoIP dostačující, proto je dalším krokem pro ověření součinnosti podpory QoS u koncového uživatele a v přístupovém bodu, nastavení klasifikace provozu a přidělování priorit v přístupovém bodu. Pro nastavení více parametrů podpory QoS v přístupovém bodu, je opět nutné se připojit do webového rozhraní adresou 192.168.1.1, a zde v nabídce *bandwidth management* viz obrázek 6.18, je nejprve zvolena kliknutím na obrázek možnost

VoIP / Video Streaming, což upřednostní multimediální datové toky, po zvolení *apply* nastává přechod do specifičtějšího nastavení priorit datového provozu. Pro přiřazení nejvyšší priority simulaci VoIP hovoru je možnost vytvořit pravidla v připravené tabulce viz obrázek 6.18 vpravo. Nejprve se přiřadí jméno pravidla např. VoIP, dále zdrojová adresa (adresa stanice PC-2 192.168.1. x), následuje číslo portu, na který je datový tok posílán *port: 5060* a v poslední řadě výše priority, což je v tomto případě *prio 1* -> *add* -> *apply* -> *save & restart*. Po restartu a připojení všech zařízení k vytvořené bezdrátové síti, probíhá opět zatížení datovým tokem jako v předchozích případech. V následujících obrázcích 6.22 a 6.23 jsou zobrazeny porovnání průběhů simulovaného VoIP hovoru s nastavenou podporou QoS v přístupovém bodu a podporou nastavenou v koncovém uživateli PC-2 vůči průběhu pouze s nastavenou podporou QoS v přístupovém bodu viz obrázek 6.23 a ve druhém případě je průběh porovnán s křivkou simulovaného VoIP hovoru zatížené sítě bez jakékoliv podpory QoS viz obrázek 6.23. Z grafů v obou obrázcích jsou viditelné rozdíly, při nastavení částečné podpory, plné podpory nebo naopak bez jakéhokoliv nastavení podpory QoS.



OBRÁZEK 6.22 - POROVNÁNÍ KŘIVEK S ÚPLNOU PODPOROU QoS(FIALOVÁ) A S ČÁSTEČNOU PODPOROU QoS(ČERNÁ)



OBRÁZEK 6.23 - POROVNÁNÍ KŘIVEK S ÚPLNOU PODPOROU QoS(FIALOVÁ) A BEZ PODPORY QoS(MODRÁ)

Černá křivka jitteru v grafu na obrázku 6.22 znázorňuje průběh, při nastavení podpory QoS využívající metodu HFSC v koncové stanici, z hodnot lze zpozorovat, že na rozdíl od modré křivky v obrázku 6.23 se tyto hodnoty pohybují v polovičním rozmezí. Modrá křivka vykresluje průběh jitteru bez nastavení jakýchkoliv parametrů podpory QoS. Naměřené výsledky ukazují, že přiřazení klasifikace provozu a priorit u koncového uživatele zvýší kvalitu provozu, ale i tak tyto hodnoty nejsou ideální. Po nastavení těchto pravidel i v přístupovém bodu jsou naměřené hodnoty dle požadavků, což lze vidět v obou obrázcích na fialových křivkách. Hodnoty jitteru jsou v případě nastavení podpory QoS do 30ms. Tento čas je pro přenos VoIP hovoru vyhovující.

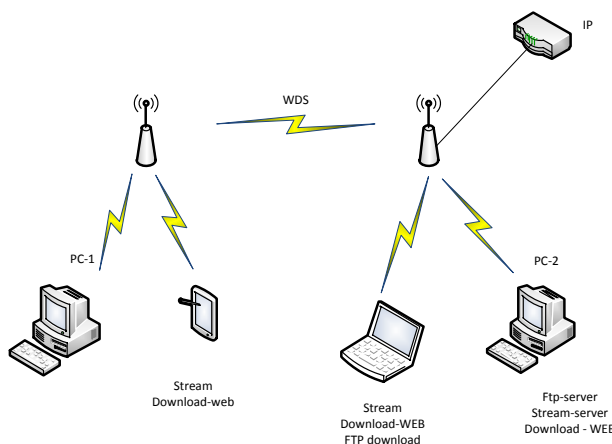
TABULKA 6.4 - DOBY ODEZVY A RYCHLOSTI FTP PŘENOSU U TOPOLOGIE 3

	bez QoS	QoS v konc. stanici	QoS v AP	QoS v AP i v konc. stanici
odezva mezi PC-1 a PC-2	1143ms	317ms	884ms	271ms
průměrná rychlost ftp	70kB/s	7kB/s	60kB/s	6kB/s

Ve výše uvedené tabulce lze porovnat hodnoty doby odezvy mezi stanicemi PC-1 a PC-2 a také průměrnou rychlost ftp přenosu v různých situacích nastavení podpory QoS. Z důvodu použití samostatného zařízení jako přístupového bodu a více stanic, lze z tabulky vidět několikanásobné zvýšení hodnoty doby odezvy a snížení rychlosti ftp přenosu. Z výsledků je zřejmé, že při využití podpory QoS v koncovém zařízení dojde k velkému snížení rychlosti ftp přenosu a zlepšení doby odezvy, což vede k zvýšení kvality přenosu multimediálních služeb.

6.4 Topologie vytvoření sítě se dvěma Access Pointy

Poslední topologie využívá dvou přístupových bodů, přičemž jeden z bodů je připojen do sítě vedené v laboratoři SC5.32 a druhý přístupový bod je propojen s přístupovým bodem pomocí technologie WDS (Wireless Distribution System) viz obrázek 6.24. V zapojení jsou využity přístupové body ASUS WL-500W.



OBRAZEK 6.24 - TOPOLOGIE SE DVĚMA PŘÍSTUPOVÝMI BODY SPOJENÝMI TECHNOLOGIÍ WDS

Pro spojení dvou přístupových bodů je třeba tyto zařízení nastavit dle následujících obrázků. Na obrázcích 6.25 a 6.26 je nastavení položky *wireless* a položek *interface a bridge* v přístupovém bodě připojenému do sítě. V tabulce *remote bridge list* je vložena MAC adresa druhého přístupového bodu.

Wireless - Interface	
SSID:	default
Channel:	9
Wireless Mode:	802.11G Only ☒ Protection
Bandwidth:	40MHz or 20MHz
Authentication Method:	Open System
WPA Encryption:	TKIP
WPA Pre-Shared Key:	
WEP Encryption:	None
Passphrase:	
WEP Key 1 (10 or 26 hex digits):	
WEP Key 2 (10 or 26 hex digits):	
WEP Key 3 (10 or 26 hex digits):	
WEP Key 4 (10 or 26 hex digits):	
Key Index:	
Network Key Rotation Interval:	0
Restore Finish Apply	

OBRÁZEK 6.25 - NASTAVENÍ AP1-1

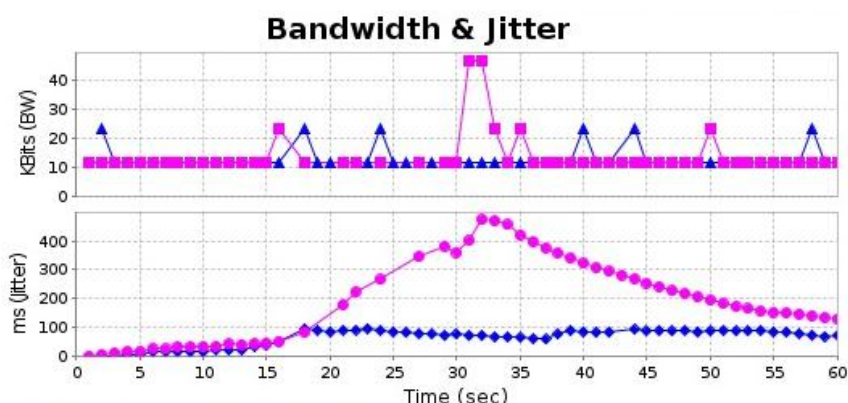
Wireless - Bridge	
Wireless bridge (also known as Wireless Distribution System or WDS) function allows you to connect to one or many APs through wireless.	
AP Mode:	Hybrid
Channel:	9
Connect to APs in Remote Bridge List?	☒ Yes ☐ No
Remote Bridge List Add Del	
MAC Address 0022152425B0	
Restore Finish Apply	
Restore:	Clear the above settings and restore the settings in effect.
Finish:	Confirm all settings and restart WL500W now.
Apply:	Confirm above settings and continue.

OBRÁZEK 6.26 - NASTAVENÍ AP1-2

Dále má tento přístupový bod nastavenou IP adresu ve složce *IP Config – WAN & LAN* na adresu 192.168.1.1 a ve složce *DHCP Server* je povolen DHCP server.

Druhý přístupový bod je nastaven podobně, v porovnání s obrázkem 6.25 je nastaveno pouze jiné SSID bezdrátové sítě a v porovnání s obrázkem 6.26 je v tabulce *Remote Bridge List* vložena MAC adresa přístupového bodu 1. MAC adresy přístupových bodů jsou k nalezení na spodní straně zařízení. Dalším rozdílem v nastavení je rozlišení IP adresy ve složce *IP Config – WAN & LAN* je nastavena adresa na 192.168.1.2 a ve složce *DHCP Server* je zakázán DHCP server, tudíž IP adresy jsou přidělovány prvním přístupovým bodem. Po nastavení obou přístupových bodů je nutno konfiguraci uložit a zařízení restartovat.

První zkouška podpory QoS je provedena využitím dvou počítačových stanic, z nichž je každá připojena k jiné bezdrátové síti, které jsou tvořeny přístupovými body. Mezi oběma počítačovými stanicemi je vytvořen datový tok jako v předchozích kapitolách. Ověření nejprve probíhá pouze s nastavením podpory QoS v přístupových bodech povolením nebo zakázáním služby WMM. Tento krok je proveden shodně s kapitolou 6.3, ještě je třeba se ujistit, že v položce *Bandwidth Management* je veškerá podpora u obou přístupových bodů vypnuta. Obrázek 6.27 ukazuje průběh šířky pásma a jitteru simulovaného VoIP hovoru v případě s vypnutou službou WMM (fialová křivka) a se zapnutou službou WMM (modrá křivka).



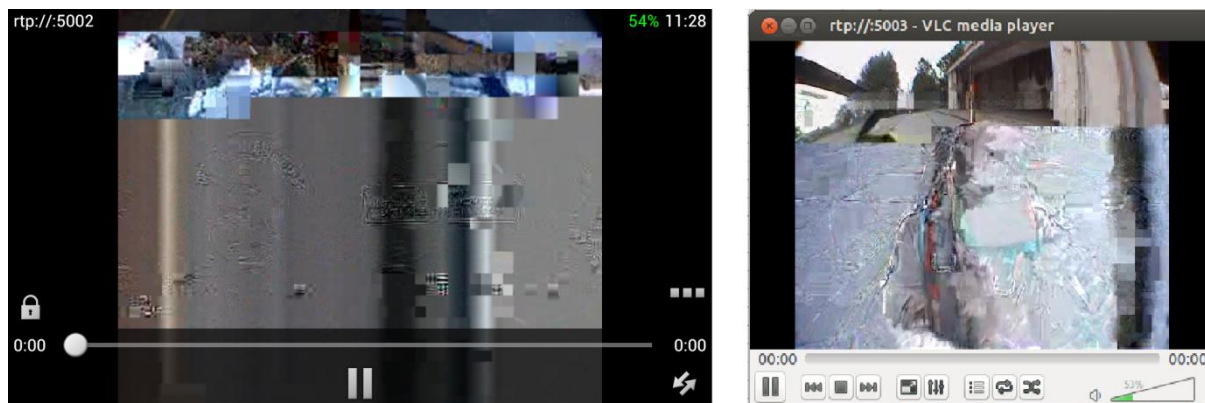
OBRÁZEK 6.27 - VYUŽITÍ SLUŽBY WMM (MODRÁ) V POROVNÁNÍ BEZ VYUŽITÍ SLUŽBY WMM (FIALOVÁ)

Grafy v obrázku 6.27 ukazují na výrazné zlepšení průběhů obou veličin, kde maximální hodnota jitteru se snížila až o 400ms z hodnoty 500ms na 100ms a šířka pásma se stabilizovala na ustálené hodnoty. Hodnota jitter 100ms, ale není dostačující pro přenos VoIP hovoru. Proto je nutné využít další metody podpory QoS nejprve v koncových stanicích s využitím metody HFSC tak následně i v obou přístupových bodech.

Další testování pro ověření funkčnosti podpory QoS v zapojení dle obrázku 6.24 již probíhá se čtyřmi připojenými zařízeními. Stanice PC-2 funguje stále jako ftp server, stanice PC-1 a notebook stahují z ftp serveru data, dále stanice PC-2 vysílá stream video do ostatních stanic. Mezi stanicemi

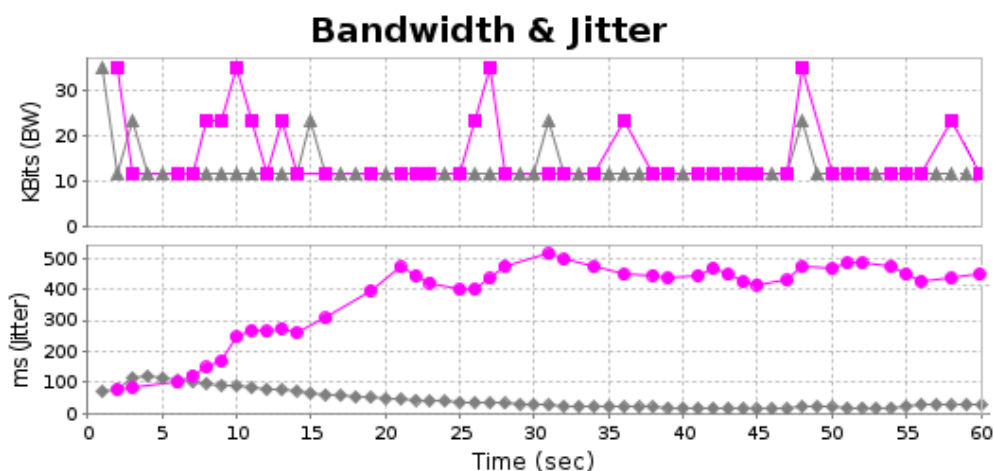
PC-1 a PC-2 také probíhá simulovaný VoIP hovor pomocí programu jperf a z počítačových stanic je prováděno měření doby odezvy mezi stanicemi pomocí programu ping.

Prvním testem je vyzkoušení topologie bez využití jakýchkoliv mechanismů pro podporu kvalitativních požadavků služeb. Testování probíhá stejným způsobem, jako v kapitole 6.3 výsledky přenášeného datového přenosu jsou podobné předchozímu zapojení. Na všech zařízeních není možné sledovat stream videa vysílaný ze stanice PC-2 viz obrázek 6.28, na kterém jsou zobrazeny výsledky přenášeného videa na mobilním telefonu a v notebooku.



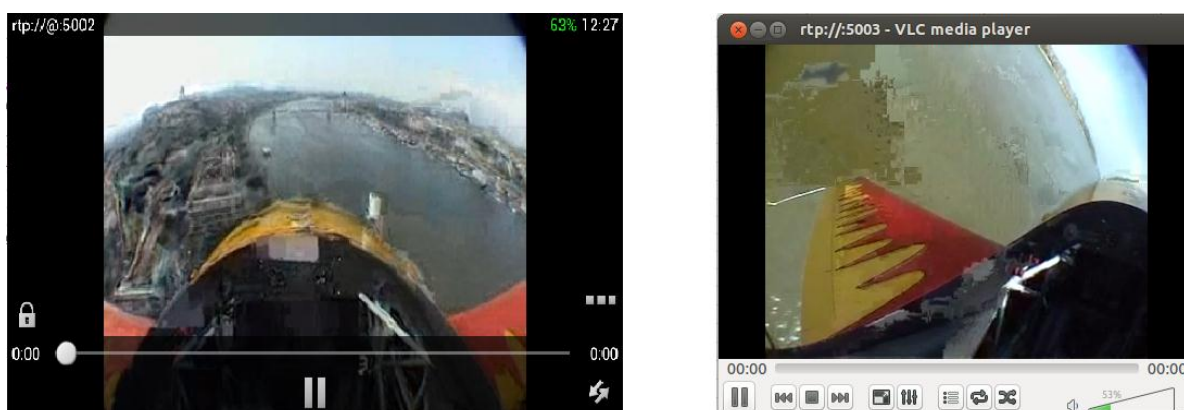
OBRÁZEK 6.28 - STREAM VIDEA V MOBILNÍM TELEFONU A NOTEBOOKU BEZ PODPORY QOS

U přenosu simulovaného VoIP hovoru se průběh jitteru pohyboval kolem hodnot 500ms a křivka šířky pásma vykazovala značně kolísavý průběh. Naměřené hodnoty dokazují potřebu použití podpory QoS v daném zapojení. Prvním způsobem upřednostnění datových toků je nastavení podpory QoS v obou přístupových bodech. V zapojení jsou použity shodné přístupové body ASUS WL-500W, tudíž nastavení probíhá shodně s předchozí kapitolou. Po nastavení obou přístupových bodů jak povolení WMM tak nastavení tabulky pro přenos VoIP hovoru a stream videa v podsložce *bandwidth management* je nutné oba přístroje restartovat. Následuje opětovné měření se stejným postupem jako v přechodném případě.



OBRÁZEK 6.29 - PRŮBĚHY S PODPOROU QOS V AP (ŠEDÁ) A BEZ PODPORY QOS (FIALOVÁ)

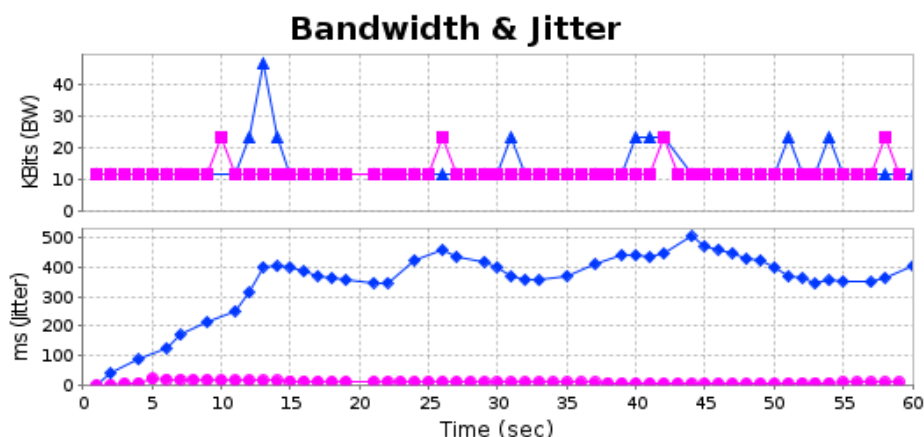
Na grafech v obrázku 6.29 jsou zobrazeny průběhy jitteru a šířky pásma v případě s nastavenou podporou upřednostnění služeb v přístupových bodech (šedá křivka) a bez jakékoliv podpory kvality služeb (fialová křivka). Výsledky ukazují vysoké zvýšení kvality přenášeného datového toku, ale v počátku se hodnota kolísání zpoždění dostává nad hodnotu 100ms, což pro přenos multimediálních služeb není žádoucí. Přehrávané přenosy videí se ve všech zařízeních podstatně zlepšily. Video jsou mírně rozmazaná a občas se pozastavují viz obrázek 6.30, který ukazuje záběr videa z mobilního telefonu a notebooku. Video na počítačové stanici PC-1 má shodný průběh s videi zobrazenými na obrázku.



OBRÁZEK 6.30 - STREAM VIDEO V MOBILNÍM TELEFONU A NOTEBOOKU S PODPOROU QOS V AP

Pro porovnání s nastavením podpory kvalitativních požadavků služeb pouze v přístupových bodech je použit postup shodný s kapitolou 6.3. Po aplikování metody HFSC do stanice PC-2 a vypnutí podpory QoS v about přístupových bodech je průběh jitteru a šířky pásma téměř shodný s předchozím nastavením, kde byla podpora QoS povolena pouze v přístupových bodech. V případě přenášeného videa je průběh takový, že video ve stanici PC-1, na kterou je vysílán stream s využitím portu 5004 probíhá plynule a video na dalších dvou stanicích je ve velmi špatné kvalitě a téměř se nepohybuje. Je to způsobeno tím, že v kódu používaném pro metodu HFSC je upřednostněno pásmo pouze pro port 5004. Tyto vlastnosti lze upravit změnou skriptu a přidělením určité šířky pásma i pro porty dalších zařízení, což je ovšem závislé na maximální rychlosti uplinku linky. Rozšíření o podporované porty se provádí stejným způsobem jako v kapitole 6.2.

Ani jedna z předchozích verzí není ideální pro přenos časově náročných služeb přenosovým médiem. Poslední variantou nastavení podpory QoS je využít možnosti nastavení podpory kvalitativních požadavků služeb jak u koncových zařízení, tak i v přístupových bodech. Zavedení podpory QoS probíhá stejně jako v předchozích bodech. Po připojení zařízení k sítím tvořeným přístupovými body opět probíhá testování funkčnosti podpory QoS v bezdrátových sítích pomocí vytvoření datového toku mezi stanicemi. Výsledek průběhu simulovaného VoIP hovoru je zobrazen na obrázku 6.31.



OBRÁZEK 6.31 - PRŮBĚHY S PLNOU PODPOROU QOS(FIALOVÁ) A BEZ PODPORY QOS (MODRÁ)

Porovnáním křivek na obrázku 6.31 a na grafech v obrázcích 6.27 a 6.29 lze zjistit, že průběh simulovaného hovoru probíhal s nejlepšími vlastnostmi v posledním případě, kde je křivka hovoru reprezentována fialovou křivkou, pro kterou jsou nastaveny parametry podpory QoS v koncových zařízeních i v přístupových bodech. Hodnoty jitteru se zde pohybují do výše asi 30ms, což je pro přenos VoIP hovoru vyhovující. Na modrých křivkách jsou zobrazeny průběhy šířky pásma a jitteru v případě bez podpory QoS. Průběh přenášeného videa opět záleží na dostupné rychlosti uplinku a na přiřazení šířky pásma použitým portům.

V tabulce 6.5 jsou naměřené hodnoty odezvy mezi stanicemi PC-1 a PC-2 a průměrné rychlosti ftp přenosu ve všech testovaných situacích. V porovnání s tabulkou 6.4 jsou hodnoty doby odezvy téměř dvojnásobné. To je způsobeno i tím, že pro spojení těchto zařízení musí projít rámec více síťovými zařízeními. Přenosové rychlosti ftp přenosu jsou v zapojení bez podpory QoS a s podporou QoS pouze v přístupových bodech podstatně nižší než tomu bylo v předchozích zapojeních, což je způsobeno opět delší přenosovou cestou. V případě s využitím metody HFSC jsou hodnoty téměř shodné a to z toho důvodu, že toto pravidlo je nastaveno již ve stanici, kde je vytvořen ftp server, tudíž ze zařízení není možné odesílat ftp data vyšší rychlostí pokud probíhá jiný přenos, který je upřednostňován.

TABULKA 6.5 - DOBY ODEZVY A RYCHLOSTI FTP PŘENOSU U TOPOLOGIE 4

	bez QoS	QoS v konc. stanici	QoS v AP	QoS v AP i v konc. stanici
odezva mezi PC-1 a PC-2	2125ms	612ms	1873ms	375ms
průměrná rychlost ftp	90kB/s	9kB/s	60kB/s	6kB/s

Kapitola 6 ověřuje, funkčnost podpory QoS v bezdrátových sítích, při nastavení této podpory jak u koncových stanic s operačními systémy unixového typu, tak s nastavením podpory QoS v samotných přístupových bodech. V kapitole je také ověřena součinnost těchto dvou metod nastavení podpory kvality služeb.

7. Laboratorní úloha

Dalším bodem zadání je navržení laboratorní úlohy pro předmět BARS – Architektura sítí. Úloha je navržena tak, aby se studenti seznámili s možnostmi nastavení podpory QoS u koncového uživatele, například pro domácí síť, ale také si ověří součinnost mezi koncovým uživatelem a přístupovým bodem. Úloha probíhá v OS Ubuntu, který je spuštěn v Oracle VM Virtualboxu. Studenti budou dané klasifikace provozu nastavovat pomocí terminálu. Kapitola stručně popisuje softwarové a hardwarové prostředky, které jsou v úloze využívány a také, postup pro přípravu laboratorního pracoviště. Laboratorní úloha je vložena v příloze.

Oracle vm virtualbox

VirtualBox je výkonný multiplatformní virtualizační stroj. Multiplatformní znamená, že je možné jej nainstalovat na Windows, Linux, Mac OS a Solaris x86. Virtualizační software umožňuje spustit více virtuálních strojů, tzn., je možné spustit více operačních systémů v jednom počítači současně, např. je možné spustit Windows i Linux na počítači Mac.

Ubuntu

Distribuce linuxu pro servery, osobní počítače a pracovní stanice, založená na jádře Debian GNU/Linux. Ubuntu znamená v překladu z jihoafrického pojmu „lidskost ostatním“, což je potvrzeno, přehledným a nesložitým grafickým prostředím. Tato distribuce je každého půl roku aktualizována s podporou na dalších 9 měsíců.

Filezilla

Filezilla je volně šiřitelná multiplatformní FTP aplikace, která poskytuje možnost FTP serveru i FTP klienta. Software podporuje FTP, SFTP, FTPS protokoly. Filezilla má přehledné uživatelské grafické rozhraní, ve kterém lze jednoduše procházet složky ftp serveru, ale také je zde zobrazena přenosová rychlost ftp přenosu a doba trvání aktuálního ftp přenosu.

Wireshark

Wireshark je volně šiřitelný paketový analyzátor. Používá se pro odstraňování síťových problémů, analýzu, vývoj softwaru a komunikačních protokolů a ke vzdělávání. Původně se program jmenoval Ethereal, ale v roce 2006 byl přejmenován na Wireshark. Je to multiplatformní program běžící ve všech operačních systémech. Poskytuje přehledné grafické rozhraní, ale k dispozici je také ve verzi pro terminál tzv. TShark.

Hostapd

Hostapd je uživatelský prostředek pro vytvoření přístupového bodu a ověření klientů v serveru. Pomocí konfiguračního souboru hostapd.conf je dostupné velké množství možností konfigurace bezdrátové sítě.

Iperf

Iperf je grafická verze síťového nástroje Iperf. Iperf je běžně používaný testovací nástroj, který je schopen vytvořit TCP a UDP datové přenosy. Program dokáže měřit propustnost sítě a jitter. Iperf umožňuje nastavit různé parametry, které mohou být použity při testování sítě, případně pro její optimalizaci a vyladění. Program podporuje funkci klienta a serveru a díky tomu umožňuje měřit propustnost mezi dvěma uživateli. Program je multiplatformní, tudíž je spustitelný ve všech operačních systémech.

VLC media player

VLC media player je multiplatformní přehrávač multimediálních služeb, který je také vybaven serverem pro streamování multimédií. Přehrávač podporuje hodně metod komprese audio a video formátů.

Ping

Nástroj pro testování počítačových sítí. Umožňuje testovat dosažitelnost stanic v IP síti, a také ověřovat dobu odezvy a ztrátovost vyslaných paketů. Ping zasílá ICMP (Internet Control Message Protocol) paket na adresu hostitele a čeká na odezvu, v rámci tohoto procesu měří dobu, která proběhne mezi odesláním ICMP paketu a následným přijetím ICMP paketu zpět.

Směrovač ASUS WL-500W

Multifunkční bezdrátový směrovač, vybavený technologií 802.11n pro vysokorychlostní bezdrátové přenosy. Dále je vybaven usb porty pro digitální stahování nebo sdílení tiskárny. Směrovač umožňuje klasifikování provozu a přiřazování priorit různým datovým tokům, což je umožněno přehledným webovým rozhraním.



OBRÁZEK 7.1 - ASUS WL-500W

USB přijímače Wi-Fi ASUS USB-N11 a Ovislink AirLive X.USB-3

Asus USB-N11 je usb příslušenství, umožňující uživateli připojení k bezdrátovým sítím. Zařízení podporuje bezdrátové standardy 802.11 b/g/n.

Ovislink Airlive X.USB-3 je přenosný USB adaptér pracující v pásmech 2.4GHz a 5GHz, který podporuje standardy 802.11 a/b/g/n. Umožňuje připojení k jakékoliv bezdrátové síti s využitím

maximální rychlosti. Adaptér je vybaven dvěma odnímatelnými anténami, což přidává možnost výměny antény v případě potřeby využití antény s vyšším zesílením.



OBRÁZEK 7.2 - ASUS USB-N11 (VLEVO), OVISLINK AIRLIVE X USB-3.0 (VPRAVO)

SmartPhone

Pro provedení úlohy je třeba jakýkoliv smartphone podporující připojení k bezdrátovým sítím wi-fi a se softwarovým vybavením VLC media player. Při provádění a navrhování laboratorní úlohy byl využit smartphone SONY Xperia S.

Příprava pracoviště

Pro provedení úlohy je třeba připravit pracoviště. Jedním z hlavních bodů přípravy je instalace virtuálního OS do počítačových stanic. Nejprve je třeba nainstalovat Oracle vm Virtualbox, který je volně dostupný na www.virtualbox.org, po instalaci virtualboxu následuje stažení volně šiřitelné distribuce linuxu Ubuntu a její nainstalování ve virtualboxu. Ubuntu je volně dostupné na www.ubuntu.com. Následně je třeba v obou stanicích nainstalovat ve virtuálním operačním systému, všechny dříve zmíněné softwarové prostředky. Ve stanici PC-2 není potřeba instalovat program Filezilla a hostapd, a to z toho důvodu, že tyto programy nejsou ve stanici PC-2 využívány. Programy lze nainstalovat pomocí příkazu v terminálu `sudo apt-get install <název programu>` nebo pomocí *centrum softwaru ubuntu*, které lze najít na liště vlevo. Ve stanici PC-2 je třeba nainstalovat ftp server. Postup instalace je dostupný na webové stránce help.ubuntu.com. V konfiguračním souboru ftp serveru je nutné nastavit uživatele, na kterého se bude stanice PC-1 přihlašovat.

Závěr

Práce je zaměřena na problematiku podpory kvalitativních požadavků služeb v bezdrátových sítích. První část pojednává o QoS obecně, jsou zde popsány parametry, na které je třeba se zaměřit při nastavení těchto služeb pro datové přenosy. Druhá část popisuje kvalitativní požadavky služeb v operačních systémech Unixového typu. Kapitola se zabývá metodami, které je možné využít pro zavedení podpory QoS, jaké jsou jejich výhody a nevýhody a jakých prvků je pro podporu QoS zapotřebí. Z popsaných metod je zřejmé, že metody HTB a HFSC jsou pro nastavení kvalitativních požadavků služeb nejvhodnější. Toto zjištění vedlo k využití metod HTB a HFSC pro otestování funkčnosti podpory QoS v bezdrátových sítích.

V šesté kapitole byly navrženy topologie a scénáře pro ověření funkčnosti těchto metod. Z naměřených výsledků, které jsou v kapitole prezentovány, je možné vyčíst změny a vliv funkčnosti použitých metod. Ze dvou vybraných metod byla vhodnější metoda HFSC a to z toho důvodu, že tato metoda umožňuje nastavení maximální doby odezvy, což je u přenosů dat v reálném čase důležitá vlastnost. Z měření v kapitolách 6.3 a 6.4 bylo zjištěno, že nastavení upřednostnění datových toků pouze v koncovém zařízení je nedostačující. Proto v tomto případě přidáváme nastavení podpory QoS v přístupových bodech, čímž docílíme zvýšené kvality požadovaného přenosu. Současně byla dokázána součinnost mezi podporou kvality přenosu služeb ve vrstvě síťové, na které pracuje nastavení metod v operačních systémech typu Unix, s vrstvou linkovou, na které pracují přístupové body. Z provedených měření bylo dokázáno, že podpora QoS na obou službách mezi sebou v bezdrátovém prostředí spolupracuje a je tedy možno tyto sítě pro přenos dat v reálném čase využít.

V poslední kapitole se práce zabývá laboratorní úlohou pro předmět Architektura sítí, která je uvedena v příloze práce. Laboratorní úloha je zaměřena na seznámení se s funkčností a metodami upřednostňování datových toků v operačních systémech Unixového typu. Úloha se nezabývá pouze podporou QoS v koncových uživateli, ale je v ní také obsažen návrh topologie s využitím přístupového bodu. Hlavním cílem použití přístupového bodu je seznámit studenty s kooperací podpory kvality přenosu dat mezi síťovou a linkovou vrstvou. V kapitole jsou stručně popsány programy a zařízení, která jsou pro provedení laboratorní úlohy potřebné. Poslední část kapitoly uvádí návod pro přípravu pracoviště před prováděním samotné laboratorní úlohy.

Seznam použité literatury:

- [1] **Bouška, Petr.** *www.samuraj-cz.com. Cisco QoS 1 - úvod do Quality of Service a DiffServ.* [Online] [Citace: 25. únor 2012.] dostupné z: <<http://www.samuraj-cz.com/>>
- [2] **Číka, ing. Petr.** *Multimediální Služby.* 30. listopad 2007.
- [3] **Wallace, Kevin.** Cisco IP Telephony Flash Cards: Weighted Random Early Detection (WRED). *informIT.* [Online] [Citace: 15. duben 2012.], dostupné z: <<http://www.informit.com/>>
- [4] **Consani, C.** Work In Progress: Quality of Service of IEEE 802.11e. [Online] 2003. [Citace: 15. únor 2012.] dostupné z: <<http://pubs.cs.uct.ac.za/>>
- [5] **Strikeback.** *A diagram of the 7-layer OSI model with the modifications made by 802.11 standard and the 802.11e amendment.* 22. květen 2009.
- [6] **Wi-Fi Alliance.** Wi-Fi CERTIFIED™ for WMM™ - Support for Multimedia Applications. [Online] 1. září 2004. [Citace: 25. duben 2012.], dostupné z: <<http://www.wi-fi.org>>
- [7] **Nagy, Andrew von.** Wireless QoS Part 4 - Arbitration Interframe Spacing. 4. Srpen 2010
- [8] **Stefan Mangold, Sunghyun Choi, Peter May, Ole Klein, Guido Hiertz, Lothar Stibor.** IEEE 802.11e Wireless LAN for Quality of Service. [Online] [Citace: 14. března 2012.], dostupné z: <<http://www2.ing.unipi.it/>>
- [9] **R.CHANDRASHEKAR, GEORGE MATHEW THARAKAN, JOSEPH THALIATH.** *IMPLEMENTATION OF HYBRID COORDINATION FUNCTION IN NETWORK SIMULATOR 2.* KATTANKULATHUR, INDIA, Květen 2006.
- [10] **Devera, Martin.** Princip a užití HTB QoS disciplíny. [Online] 2002. dostupné z <<http://luxik.cdi.cz/~devik/qos/htb/cs/slt02htb.ps>>
- [11] **Brown, Martin A.** Traffic Control HOWTO. [Online] 2006. dostupné z: <<http://tldp.org/HOWTO/Traffic-Control-HOWTO/index.html>>.
- [12] Classful Queueing Disciplines. *Linux Advanced Routing & Traffic Control HOWTO.* [Online] dostupné z: <<http://lartc.org/howto/lartc.qdisc.classful.html>>.
- [13] **Klaus Rechart, Patrick McHardy.** HFSC Scheduling with Linux. [Online] 2005. dostupné z: <<http://linux-ip.net/articles/hfsc.en/>>.
- [14] **Ion Stoica, Hui Zhang, T. S. Eugene Ng.** A Hierarchical Fair Service Curve Algorithm for Link-Sharing, Real-Time and. *sigcomm.* [Online] 1997. dostupné z: <<http://conferences.sigcomm.org/sigcomm/1997/papers/p011.pdf>>

Seznam obrázků:

Obrázek 2.1 - dělení služeb QoS z hlediska komplexnosti	13
Obrázek 3.1 - vrstvý model OSI s modifikacemi 802.11 a 802.11e [5]	15
Obrázek 3.2 - WMM doba přístupu k médiu u jednotlivých tříd přenosu	17
Obrázek 3.3 - DCF Okamžitý přístup pro první rámeček, když je médium volné (DIFS/AIFS[i])	17
Obrázek 3.4 - přístup k médiu EDCF	19
Obrázek 3.5 - osm přenosových kategorií (TC), vlevo DCF AIFS=34us CWmin=15, PF=2; vpravo EDCF AIFS[TC]>=34us, CWmin[TC]=0-255, PF[TC]=1-16 [8].	19
Obrázek 3.6 - typický super rámeček u 802.11e [9].....	20
Obrázek 4.1 – hierarchické sdílení linky	21
Obrázek 5.1 – základní řazení do front FIFO	28
Obrázek 5.2 – výchozí qdisc v Os unixového typu PFIFO_FAST.....	29
Obrázek 5.3 – SFQ metoda pro spravedlivé dělení dat do front.....	29
Obrázek 5.4 – metoda pro tvarování datového toku token bucket filter - TBF	30
Obrázek 5.5 - příklad PRIO.....	32
Obrázek 5.6 - příklad CBQ	35
Obrázek 5.7 – příklad vypůjčování tokenů	37
Obrázek 5.8 - hierarchické sdílení přenosové kapacity.....	40
Obrázek 5.9 - vývojový diagram aktualizace virtuálního času.....	44
Obrázek 6.1- topologie se dvěma stanicemi, z nichž jedna slouží jako AP	45
Obrázek 6.2- stream bez QoS vlevo nezatížená síť vpravo zatížená síť.....	46
Obrázek 6.3 - šířka pásma a jitteru u simulovaného VoIP bez zatížení	46
Obrázek 6.4 –šířka pásma a jitteru u simulovaného VoIP se zatížením	47
Obrázek 6.5 - rozdělení šířky pásma HTB	47
Obrázek 6.6 - výpis nastavení HTB	48
Obrázek 6.7 - rozdělení HTB	49
Obrázek 6.8 - stream videa ze zapojení s nastavenými prioritami HTB	49

Obrázek 6.9 - šířka pásma a jitter s nastavnou podporou QoS pomocí HTB.....	50
Obrázek 6.10 - výpis metody HFSC.....	51
Obrázek 6.11 - rozdělení HFSC	52
Obrázek 6.12- šířka pásma a jitter s HFSC(zelená) a bez HFSC (fialová).....	52
Obrázek 6.13 - topologie se třemi stanicemi, z nichž jedna slouží jako AP	53
Obrázek 6.14 - stream videa na mobilním telefonu bez podpory QoS	54
Obrázek 6.15 - stream videa na mobilní telefon s opdporou QoS	55
Obrázek 6.16 - stream videa na stanici PC-1	55
Obrázek 6.17 - topologie s přístupovým bodem	56
Obrázek 6.18 - možnosti rozdělení šířky pásma Asus WL-500W	57
Obrázek 6.19 – simulovaný VoIP s metodou HFSC ve stanici PC-2(fialová) a bez metody HFSC(šedá)	58
Obrázek 6.20 - povolení metody WMM.....	58
Obrázek 6.21 - simulovaný VoIP s povolenou službou WMM s(žlutá) a bez metody HFSC(fialová).....	59
Obrázek 6.22 - porovnání křivek s úplnou podporou QoS(fialová) a s částečnou podporou QoS(černá)	60
Obrázek 6.23 - porovnání křivek s úplnou podporou QoS(fialová) a bez podpory QoS(modrá)	60
Obrázek 6.24 - topologie se dvěma přístupovými body spojenými technologií WDS	61
Obrázek 6.25 - nastavení AP1-1	62
Obrázek 6.26 - nastavení AP1-2	62
Obrázek 6.27 - využití služby WMM (modrá) v porovnání bez využití služby WMM (fialová).....	63
Obrázek 6.28 - stream videa v mobilním telefonu a notebooku bez podpory QoS.....	64
Obrázek 6.29 - průběhy s podporou QoS v AP (šedá) a bez podpory QoS (fialová).....	64
Obrázek 6.30 – stream videa v mobilním telefonu a notebooku s podporou QoS v AP	65
Obrázek 6.31 – srůběhy s plnou podporou QoS(fialová) a bez podpory QoS (modrá)	66
Obrázek 7.1 - Asus WL-500W	68
Obrázek 7.2 - Asus USB-n11 (vlevo), Ovislink Airlive x USB-3.0 (vpravo).....	69

Seznam tabulek:

Tabulka 3.1- kategorie přístupu WMM	16
Tabulka 3.2 - časy přístupu k médiu u různých standardů [7]	18
Tabulka 3.3 - porovnání mezirámcových mezer a CWmin, CWmax.....	18
Tabulka 5.1 - HTB – půjčování tokenů	36
Tabulka 6.1 – hodnoty naměřené v zapojení 1 bez podpory QoS	47
Tabulka 6.2 – doby odezvy a rychlosti ftp přenosu u topologie 1.....	52
Tabulka 6.3 - doby odezvy a rychlosti ftp přenosu u topologie 2	56
Tabulka 6.4 - doby odezvy a rychlosti ftp přenosu u topologie 3	61
Tabulka 6.5 - doby odezvy a rychlosti ftp přenosu u topologie 4	66

Seznam rovnic:

Rovnice 3.1 - příklad výpočtu doby trvání přístupu k médiu pro udělenou prioritu hlasu	18
Rovnice 5.1 - rovnice křivky termínu $D_i()$	40
Rovnice 5.2 - křivka časové náročnosti $E(t)$	41
Rovnice 5.3 - nejhorší případ kdy se všechny pasivní relace stanou aktivními	41
Rovnice 5.4 - první krok odvození křivky časové náročnosti.....	42
Rovnice 5.5 - druhý krok odvození křivky časové náročnosti.....	42
Rovnice 5.6 - finální definice křivky časové náročnosti.....	42
Rovnice 5.7 - definice virtuální křivky.....	44

Seznam příloh:

Příloha 1 – laboratorní Úloha – Architektura Sítí	77
Příloha 2 - CD	

Seznam zkratek:

AC	-	access categories
AIFS	-	Arbitration Inter-Frame Space
CBQ	-	Clase Based Queueing
CFP	-	Contention Free Period
CIR	-	Committed Information Rate
CP	-	Contention Period
CSMA/CA	-	Carrier Sense Multiple Avoidence with Colision Avoidance
CW	-	Contention Window
DCF	-	Distributed Coordination Function
DiffServ	-	Differentiated Services
DiFS	-	DCF inter-frame Space
DSCP	-	Differentiated Services Code Point
EDCF	-	Enhanced Distributed Coordination Function
ESFQ	-	Extended Stochastic Fair Queuing
EWMA	-	Exponentially Weighted Moving Average
FIFO	-	First In First Out
GRED	-	Generic Random Early Drop
HC	-	Hybrid Coordinator
HCF	-	Hybrid Coordination Function
HFSC	-	Hierarchical Fair Service Curve
HPFQ	-	Hierarchical Packet Fair Queueing
HTB	-	Hierarchical Token Bucket
ICMP	-	Internet Control Message Protocol
IEEE	-	The Institute of Electrical and Electronics Engineers
IntServ	-	Integrated Services
IP	-	Internet Protocol
LLC	-	Logical Link Control
MAC	-	Media Access Control
Mpu	-	Minimum Packet unit
MSDU	-	MAC Service Data Unit
OS	-	Operating System
PC	-	Point Coordinator

PCF	-	Point Coordination Function
PF	-	Persistence Factor
PFQ	-	Packet Fair Queueing
PIFS	-	PCF Inter-Frame Space
PPP	-	Point to Point Protocol
PPTP	-	Point to Point Tunneling Protocol
Qdisc	-	Queueing Discipline
QoS	-	Quality of Service
SEFF	-	Smallest Eligible Finish time First
SFF	-	Smallest Finish time Firsts
SFQ	-	Stochastic Fair Queueing
SSF	-	Smallest Start time First
TBF	-	Token Bucket Filter
TC	-	Traffic Categories
TCP	-	Transmission Control Protocol
ToS	-	Type of Service
TXOP	-	Transmission Opportunity
VoIP	-	Voice over IP
WDS	-	Wireless Distribution System
WiFi	-	Wireless Fidelity
WLAN	-	Wireless Local Area Network
WMM	-	Wireless MultiMedia
WRR	-	Weighted Round Robin

Podpora kvality služeb v OS Unixového typu pro provoz v bezdrátových sítích

Cíl

Cílem úlohy je seznámit se s využitím podpory kvalitativních požadavků služeb (QoS) při přenosech dat bezdrátovou sítí Wi-Fi v operačních systémech Unixového typu. Dále také seznámení s nastavením podpory QoS u koncových uživatelů používajících některou z distribucí Linuxu a také vytvoření datového toku pro zatížení sítě pomocí FTP přenosu, streamování videa a http provozu.

Vybavení pracoviště

2x PC, software: Oracle vm virtualbox, Ubuntu, Filezilla, wireshark, hostapd, jperf, VLC media player, směrovač ASUS WL-500W, USB přijímače Wi-Fi ASUS USB-N11 a Ovislink AirLive X.USB-3, smartphone (Sony XPERIA S).

Úkoly

1. Zprovozněte v koncových stanicích pomocí Virtualboxu OS Ubuntu a v něm potřebné aplikace pro generování/příjem provozu.
2. Vytvořte datový provoz mezi stanicemi, zobrazte samotné přenosy a změřte jejich kvalitu pomocí doporučeného softwaru. Vyhodnoťte kvalitu přenášeného videa.
3. Nastavte parametry QoS v koncových stanicích a spusťte opět datový provoz pro zatížení.
4. Změřte nyní kvalitu přenosu a porovnejte s původními hodnotami a snímky z přenosu bez nastavení podpory QoS.
5. V případě třetí topologie porovnejte součinnost podpory QoS při nastavení v koncových uživatelských a v přístupovém bodě.

Teoretický úvod

Quality of Service je termín, který se používá pro technologie řídicí datový tok v sítích dle požadavků služeb v sítích na bázi technologií s přepojováním paketů. QoS musí umět rozlišit druh přenosu, aby pomocí této informace mohl nastavit požadovanou prioritu danému typu datového přenosu. Priorita pro daný typ přenosu udává to, že se pakety důležitějšího typu přenosu dostanou do stanice příjemce ve správnou dobu a nepoškozeny.

Nejvíce žádoucí nastavení QoS je u přenosů v reálném čase, tzv. Real-time přenosů. U přenosů Real-time je třeba mít nastaveny priority tak, aby při přenosu dat nedocházelo ke ztrátě a zpoždění dat.

Real-time služby nepožadují pouze bezchybný přenos dat, ale hlavně přenos dat v požadované časové relaci. Nejdelší hranice pro přenos dat se nazývá deadline. U přenosů v reálném čase je možno ztratit některé pakety, ale je nežádoucí, aby pakety došly v jiném pořadí a se zpožděním.

Parametry, které se pro podporu QoS nastavují, jsou zpoždění (delay), které udává dobu, za kterou data dorazí od vysílací stanice k cílové. Dále jitter (kolísání) tato hodnota udává rozdílné zpoždění mezi

pakety. Ztrátovost (packet loss) – kolik paketů se při přenosu ztratí. Propustnost (throughput) – maximální zátěž linky, doručení mimo pořadí (out of delivery) a šířka pásma (bandwidth).

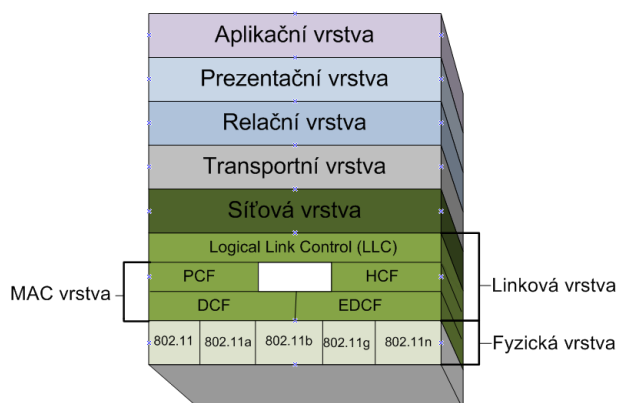
Pro přenosy v reálném čase existují doporučené hodnoty těchto parametrů pro plynulý přenos hlasu nebo videa. Příklad doporučených hodnot parametrů QoS pro internetovou telefonii VoIP:

- Latence (doba mezi vysláním a doručením paketu) < 150 ms.
- Jitter (rozdíl intervalů přijímaných paketů) < 30 ms.
- Packet loss (podíl přijatých a poslaných paketů za čas) > 99%.
- Bandwidth (souvisí s propustností) 12 – 106 kbit/s v závislosti na vzorkování, kodeku a režii na 2. vrstvě.

Následující tabulka zobrazuje dělení priorit podle kategorie datového toku:

třída přístupu	popis	výška priority dle 802.1D
Priorita Hlasu	Nejvyšší priorita: umožní více souběžných hovorů VoIP s nízkým zpožděním a vysokou kvalitou zvuku	7,6
Priorita Video	Upřednostní přenos videa před ostatními datovými toky. Jeden přístupový kanál v síti 802.11 g nebo 802.11a může podporovat 3-4 SDTV streamy nebo jeden HDTV stream.	5,4
Priorita Best Effort	Přenos z ostatních zařízení nebo aplikací, které nevyžadují QoS. Přenosy s nižší citlivostí na zpoždění. Např. internetový prohlížeč.	0,3
Ostatní priority na pozadí (background)	Nejnižší priorita: tiskové úlohy, přenos dat, přenosy, které nemají přísné požadavky na zpoždění a propustnost	1,2

Pro podporu kvalitativních požadavků služeb v bezdrátových sítích existuje standard IEEE 802.11e, který přináší možnost prioritizace přenosu také v bezdrátových sítích. Standard 802.11e zavádí nové režimy vrstvy MAC poskytující podporu QoS. Metody se nazývají EDCF (Enhanced Distributed Coordination Function) a HCF (Hybrid Coordination Function) viz obrázek 0.1



OBRÁZEK 0.1 - VRSTVOVÝ MODEL OSI S MODIFIKACEMI 802.11 A 802.11E

V operačních systémech Unixového typu je podpora QoS řešena pomocí front. Fronty paketů (queueing discipline = qdisc) slouží určení dat, která budou odeslána jako první. Příchozí provoz je označován jako ingress a jeho tvarování (přeuspořádání, zpomalení, či zahození paketu) je nazýváno policing. V OS Linux je pro ingress používáno pouze omezení šířky pásma. Odchozí provoz označován jako egress využívá ke tvarování tzv. shaping. Shapingem lze pakety zpomalovat, přeuspořádávat i zahazovat za použití různě složitých algoritmů.

Classless fronty, jsou fronty, které řídí základní typy provozu pomocí přeuspořádání, zpomalení a zahození paketu. Tento typ qdisc neumožňuje vytvoření tříd. Classless qdisc: fifo (first in first out), pfifo_fast, sfq (stochastic fair queueing), esfq (extended stochastic fair queueing), gred (generic random early drop), tbw (token bucket filter).

Classfull třídy mohou obsahovat další classless i classful qdisc. Tímto tvoří strom tříd skládaný z uzlů obsahujících další classful qdisc a listů obsahujících jeden classless qdisc. Každé síťové rozhraní má na výstupu jednu kořenovou disciplínu qdisc nastavenou do výchozího nastavení pfifo_fast, která je používána pro tvarování odchozího směru. O vstup na rozhraní se stará policing, který má pouze omezené možnosti nastavení. Classfull qdisc :prio, cbq (class based queueing), HTB (hierarchical token bucket), HFSC (hierarchical fair service curve).

Nejpoužívanějšími a zároveň nejvhodnějšími metodami jsou HTB a HFSC.

Metoda HTB

HTB – Hierarchical Token Bucket – využívá konceptu token a bucket spojeného se systémem založeném na třídách a filtrech, pro povolení komplexní kontroly nad přenosem. S komplexním modelem půjčování, může HTB provádět celou řadu sofistikovaných technik kontroly přenosu. Jednou z nejjednodušších metod použití HTB je použití tvarování.

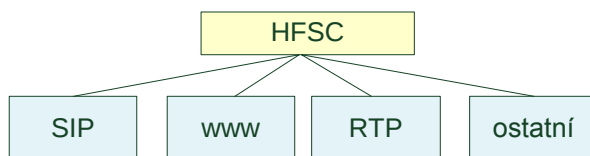
Pro pochopení metody HTB, je důležitým logickým krokem chápat metodu token bucket a funkci TBF. Tato disciplína front umožňuje uživateli definovat charakteristiku metody token a bucket a povoluje vkládat do bucketu tokeny svévolně. Při spojení metody se schématem klasifikace, může být přenos kontrolován velice podrobným způsobem.

Metoda HFSC

HFSC – Hierarchical Fair Service Curve se blíží servisní křivce modelu sdílení linky. Křivka zde definuje požadavky QoS třídy datového toku. Plánování je zde založeno na dvou kritériích: real-time kritéria, které zajišťují garanci služeb všech listových tříd, a kritéria sdílení linky (link-sharing), která si klade za cíl uspokojit servisní křivku vnitřních tříd a spravedlivě rozdělit přebytečnou šířku pásma. Kritéria real-time jsou používána k výběru paketů, pouze pokud hrozí potenciální nebezpečí, že garance služeb pro uzly typu list jsou v ohrožení. V opačném případě jsou využity kritéria sdílení linky. Tato politika zajišťuje garanci v reálném čase třídám typu list, zatímco ve stejném čase se minimalizuje rozdíl mezi aktuálními službami přijímanými do vnitřních uzlů a ty jsou řízeny modelem sdílení linky.

Následující část zobrazuje možnost rozdělení šířky pásma 1Mbit do tříd podle obrázku 0.2. Příklad rozdělení datového toku, přiřazuje protokolu SIP 100 kbit a maximální zpoždění 30ms, které je požadováno, hodnota ul rate udává celkovou šířku pásma uploadu, kterou je možno využít. Z tohoto

důvodu je u všech tříd shodná, protože v případě nevyužití šířky pásma některými třídami, je možno tuto šířku využít třídou jinou. Pro stream videa RTP je zde maximální šířka pásma 700kbit a maximální doba zpoždění 200ms, což je podle tabulek dostačující. Pro http a ostatní služby je rozdělena zbylá šířka pásma a není nastavena maximální doba zpoždění. Pomocí filtrů a v nich využitého klasifikátoru u32, je datový tok klasifikován do jednotlivých tříd. A Poslední část přiřazuje filtry k jednotlivým datovým tokům.



OBRÁZEK 0.2 - ROZDĚLENÍ ŠÍŘKY PÁSMY

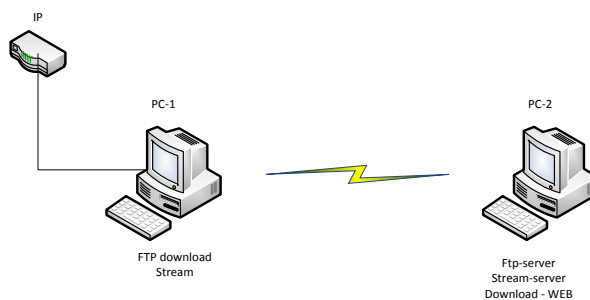
```

tc qdisc add dev wlan0 root handle 1: hfsc default 13
tc class add dev wlan0 parent 1: classid 1:1 hfsc sc rate 1000kbit ul rate 1000kbit
tc class add dev wlan0 parent 1:1 classid 1:10 hfsc sc umax 1500b dmax 30ms rate 100kbit ul rate 1000kbit
tc class add dev wlan0 parent 1:1 classid 1:11 hfsc sc umax 1500b dmax 200ms rate 700kbit ul rate 1000kbit
tc class add dev wlan0 parent 1:1 classid 1:12 hfsc sc rate 100kbit ul rate 1000kbit
tc class add dev wlan0 parent 1:1 classid 1:13 hfsc sc rate 100kbit ul rate 1000kbit
tc filter add dev wlan0 protocol ip parent 1:0 prio 1 u32 match ip sport 5060 0xffff flowid 1:10
tc filter add dev wlan0 protocol ip parent 1:0 prio 1 u32 match ip dport 5060 0xffff flowid 1:10
tc filter add dev wlan0 protocol ip parent 1:0 prio 1 u32 match ip sport 5004 0xffff flowid 1:11
tc filter add dev wlan0 protocol ip parent 1:0 prio 1 u32 match ip dport 5004 0xffff flowid 1:11
tc filter add dev wlan0 protocol ip parent 1:0 prio 1 u32 match ip sport 80 0xffff flowid 1:12
tc filter add dev wlan0 protocol ip parent 1:0 prio 1 u32 match ip dport 80 0xffff flowid 1:12
tc qdisc add dev wlan0 parent 1:10 handle 30: sfq perturb 10
tc qdisc add dev wlan0 parent 1:11 handle 40: sfq perturb 10
tc qdisc add dev wlan0 parent 1:12 handle 50: sfq perturb 10
tc qdisc add dev wlan0 parent 1:13 handle 60: sfq perturb 10
  
```

Postup řešení

Zapojení 1:

- Vytvořte zapojení dle obrázku 0.3 s využitím USB přijímače Ovislink AirLive X.USB-3 ve stanici PC-1 a Wi-Fi ASUS USB-N11 ve stanici PC-2.

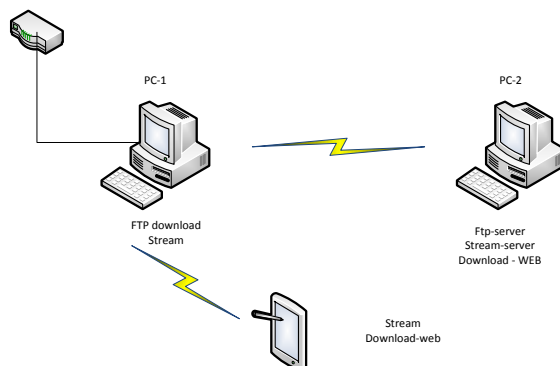


OBRÁZEK 0.3 - SCHÉMA ZAPOJENÍ 1

1. Spusťte na obou stanicích ve virtuálním stroji, operační systém UBUNTU. V těchto stanicích použijte přiložených USB adaptérů a vytvořte bezdrátovou síť. Pro povolení adaptérů ve wm: vrchní nabídka *zařízení* -> *usb zařízení* -> *povolit usb adaptér*. Nastavení ve stanici PC-1 *Nabídka připojení vpravo nahoře* -> *create wireless network* -> *název sítě, zabezpečení sítě*. Na stanici PC-2 se k této síti připojte.
2. Vytvořte datový provoz mezi těmito stanicemi. Pomocí VLC media playeru spusťte stream videa na stanici PC-2: *media* -> *stream* -> *add* -> *stream* -> *next* -> *RTP/MPEG Transport Stream* -> *add* -> *<ip adresa pc-1>* *port např. 5004* -> *next* -> *stream*. Na stanici PC-1 tento síťový stream spusťte pomocí VLC media playeru: *media*-> *open network stream* -> *rtp://@:port* -> *play*.
V další řadě otevřete ve stanici PC-1 program FileZilla a spusťte stahování z ftp serveru na stanici PC-2. Postup: *Host <ip adresa PC-2>* -> *username: anonymous* -> *password: anonymous*-> *quick connect*->*kopírovat nějaký soubor*. V další řadě spusťte v obou stanicích přes terminál program JPerf pro simulaci VoIP hovoru. *Terminal* -> *cd <cesta do složky JPerf> (/home /user/Downloads/ JPerf/)* -> *sudo sh jperf.sh*. Stanici PC-1 nastavte jako server pro port 5060 a stanici PC-2 jako client. Použijte UDP přenos se šířkou pásma 12,5 kbytes. Délku přenosu zvolte sami (minimálně 60s). Dále ve stanici PC-2 začněte stahovat soubor z internetu a spusťte ping z jedné stanice na druhou. Pozorujte a zaznamenejte si výsledky přenosu streamovaného videa, křivku jitteru, rychlost ftp přenosu, pingu a downloadu.
3. Nyní přiřadte ve stanici PC-2 priority uploadu datovým tokům pomocí metody HFCS a poté přiřadte tuto prioritu i ve stanici PC-1 pro toky, které jdou ze stanice PC-1 do stanice PC-2. Rozdělte datový tok do různých tříd, klasifikujte tyto datové toky a přiřadte jim šířky pásma tak, aby hlavně multimediální přenos pomocí stramu a hovor VoIP probíhaly beze ztrát a s minimálním zpožděním. Postup: *terminal* -> *tc qdisc add dev (wlan0, ...) root handle 1: hfsc default 12 (záleží na rozdělení datového toku)* -> *více viz teoretický úvod*. Ověřte si rozdělení datového toku do tříd pomocí příkazu *tc -s -d class show dev wlan0*.
4. Opět spusťte datový provoz mezi stanicemi a porovnejte výsledky (ftp, ping, ...) s výsledky naměřenými předtím. Dle těchto výsledků změňte nastavené HFSC hodnoty tak, aby probíhaly přenosy v reálném čase bez jakéhokoliv poškození. Navržené hodnoty zaznamenejte.

Zapojení 2:

- Vytvořte zapojení dle obrázku 0.4.

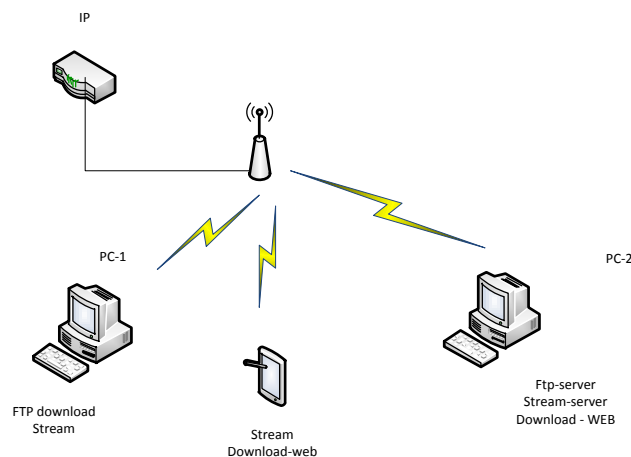


OBRAZEK 0.4 - SCHÉMA ZAPOJENÍ 2

1. Nechejte vytvořenu bezdrátovou síť z úkolu č. 1. Ve stanici PC-1 je třeba spustit authenticator pro připojení mobilního telefonu k síti. Postup: *terminal -> sudo apt-get install hostapd*. Nejprve je třeba nastavit konfigurační soubor programu: *terminal -> cd /etc/hostapd/ -> nano (vi, gedit, ...) hostapd.conf -> nastavení dle sítě vytvořené stanicí PC-1 (SSID, interface, password...) -> ctrl+x (uložení) -> sudo hostapd hostapd.conf*. Nyní je možné se připojit s mobilním telefonem k bezdrátové síti vytvořené stanicí PC-1.
2. Nyní v programu VLC běžícím na stanici PC-2 změňte stream ne pouze na stanici PC-1 ale přidejte i stream pro mobilní telefon s nainstalovaným VLC media playerem. Postup je shodný jako v předchozím případě pouze dát 2x *RTP/MPEG Transport Stream -> add -> <ip adresa zařízení> port*, pro každé zařízení zvolte jiný port (např. 5002 a 5004). Spusťte download v mobilním telefonu a poté spusťte plný datový provoz jako v úkolu č. 1. Všimněte si výsledků v počítačové stanici a v mobilním telefonu.
3. Změňte nastavení metody HFSC v obou PC stanicích tak, aby bylo přenášené video plynulé na mobilním telefonu. To znamená, změňte nastavené hodnoty pro priority přenosu streamovaného videa na stanici PC-1 tak, aby náleželi mobilnímu telefonu (porty). Změněné hodnoty zaznamenejte.
4. Rozdělte datový provoz ve stanicích PC-1 a PC-2 tak, aby byl přenos streamovaného videa plynulý v PC-1 i v mobilním zařízení a také aby přenos simulovaného VoIP hovoru probíhal dle požadavků.

Zapojení 3:

- Vytvořte zapojení dle obrázku 0.5.



OBRÁZEK 0.5 - SCHÉMA ZAPOJENÍ 3

1. Připojte do sítě směrovač ASUS WL-500W a nastavte jej jako AP. Připojte směrovač pomocí kabelu k jedné z PC stanic a připojte se k webovému rozhraní směrovače pomocí adresy 192.168.1.1. Ve složce wireless a podsložce bridge nastavte AP only. Ve složce Bandwidth Management se ujistěte, že je veškerá podpora QoS vypnuta. Stiskněte finish pro uložení a restartování AP.
2. Všechny stanice připojte k bezdrátové síti tvořené AP a vytvořte mezi nimi různý datový provoz jako v předchozích úkolech. Zaznamenejte a porovnejte výsledky s předchozími úkoly.

3. V počítačových stanicích vytvořte pomocí metody HFSC rozdělení datového toku a přiřazení filtrů. Zjistěte, jaký má vliv toto nastavení na přenos, a vyhodnoťte výsledky.
4. Nyní se připojte opět k webovému rozhraní směrovače a povolte v něm ve složce Bandwidth Management podporu QoS při nejmenším pro VoIP a stream. Vytvořte opět zatížení sítě a vyhodnoťte rozdíly mezi předchozím přenosem a současným přenosem.
5. Vytvořte a nastavte vhodné nastavení parametrů QoS pro přenos obsahující multimediální služby tak, aby tyto služby přenášeny v reálném čase nebyly nijak poškozeny. Vhodné nastavení vytvořte jak v PC stanicích tak i v přístupovém bodu.

Kontrolní otázky

1. Jaké metody přístupu k médiu jsou používány u standardu 802.11e?
2. Na které vrstvě modelu OSI a ve které její části jsou metody přístupu k médiu u standardu 802.11e definovány?
3. Jaký je rozdíl mezi classless qdisc a classfull qdisc?
4. Čeho využívá metoda HFSC pro řízení a tvarování datového toku?