

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ**

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

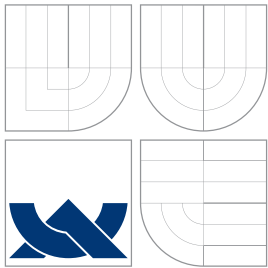
**AUTOMATICKÉ UKLÁDÁNÍ DIGITÁLNÍHO SATELITNÍHO
VYSÍLÁNÍ**

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

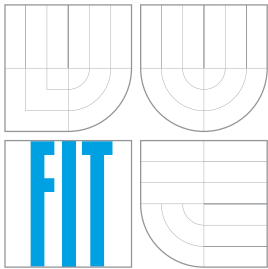
AUTOR PRÁCE
AUTHOR

Bc. IVO ŘEZNÍČEK

BRNO 2007



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

AUTOMATICKÉ UKLÁDÁNÍ DIGITÁLNÍHO SATELITNÍHO VYSÍLÁNÍ

AUDIOVISUAL RECORDING SYSTEM

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. IVO ŘEZNÍČEK

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. IGOR SZÖKE

BRNO 2007

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií

Akademický rok 2006/2007

Zadání diplomové práce

Řešitel: **Řezníček Ivo, Bc.**

Obor: Počítačová grafika a multimédia

Téma: **Automatické ukládání digitálního satelitního vysílání**

Kategorie: Softwarové inženýrství

Pokyny:

1. Prostudujte normy digitálního přenosu signálu ze satelitu (DVB-S), rozhraní digitální satelitní karty SkyStar2, rozhraní DiSEqC.
2. Navrňte aplikaci pro automatické nahrávání digitálního satelitního vysílání (zaměřte se zejména na audio a teletextovou složku) pro operační systém Linux.
3. Implementujte aplikaci.
4. Ověřte její funkčnost (ve spolupráci se speech@FIT).
5. Zhodnoťte dosažené výsledky a navrhněte případná další možná rozšíření aplikace.

Literatura:

- Normy DVB
- Dle pokynů školitele

Při obhajobě semestrální části diplomového projektu je požadováno:

- Body 1, 2 a alespoň část bodu 3 ze zadání.

Podrobné závazné pokyny pro vypracování diplomové práce naleznete na adrese <http://www.fit.vutbr.cz/info/szz/>

Technická zpráva diplomové práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap, které byly vyřešeny v rámci ročníkového a semestrálního projektu (30 až 40% celkového rozsahu technické zprávy).

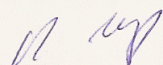
Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním paměťovém médiu (disketa, CD-ROM), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Szöke Igor, Ing.**, UPGM FIT VUT

Datum zadání: 28. února 2006

Datum odevzdání: 22. května 2007

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií
612 66 Brno, Sozotěchova 2



doc. Dr. Ing. Pavel Zemčík
vedoucí ústavu

LICENČNÍ SMLOUVA
POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami

1. Pan

Jméno a příjmení: **Bc. Ivo Řezníček**

Id studenta: 49331

Bytem: Sudice 130, 680 01 Sudice

Narozen: 14. 12. 1982, Boskovice

(dále jen "autor")

a

2. Vysoké učení technické v Brně

Fakulta informačních technologií

se sídlem Božetěchova 2/1, 612 66 Brno, IČO 00216305

jejímž jménem jedná na základě písemného pověření děkanem fakulty:

.....

(dále jen "nabyvatel")

Článek 1

Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):
diplomová práce

Název VŠKP: Automatické ukládání digitálního satelitního vysílání

Vedoucí/školitel VŠKP: Szöke Igor, Ing.

Ústav: Ústav počítačové grafiky a multimédií

Datum obhajoby VŠKP:

VŠKP odevzdal autor nabyvateli v:

tištěné formě počet exemplářů: 1

elektronické formě počet exemplářů: 2 (1 ve skladu dokumentů, 1 na CD)

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

Článek 2 Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti:
 - ihned po uzavření této smlouvy
 - 1 rok po uzavření této smlouvy
 - 3 roky po uzavření této smlouvy
 - 5 let po uzavření této smlouvy
 - 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3 Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne:

.....

Nabyvatel

.....

Autor

Abstrakt

Tato práce má za cíl vytvoření systému pro masivní nahrávání multimediálních dat. Pořizovaná data budou řečové nahrávky v co největším množství různých jazyků. Jako vhodný zdroj dat bylo zvoleno digitální satelitní vysílání (DVB-S). Tato práce se zabývá realizací systému pro správu a ukládání řečových nahrávek v digitální podobě. Systém má následující vlastnosti: paralelní nahrávání více programů, podpora více satelitních karet, podpora více parabol (DiSEqC), získávání a ukládání doplňujících informací (z internetu) jako je například jazyk vysílání. Data systémem nashromážděná budou použita k trénování systému pro identifikaci jazyka.

Klíčová slova

DVB, DVB-S, DVB-S2, MPEG 2, MPEG 1, Transport stream, Program stream, DiSEqC, Skystar 2, audio, MP2, titulky, teletext, transpondér

Abstract

This work aims at the creation of a system of massive recording of multimedia data, especially speech data in various languages. The first issue is to find out high quality data source, the second is to build the system for managing and storing received data in the digital form. A digital satellite transmission is chosen as a signal source (DVB-S system). Main system features include recording of multiple streams in parallel, support of multiple cards, retrieving and storing of additional information (from Internet) and scheduling of recordings. The system will provide massive amounts of data for training of a language identification system.

Keywords

DVB, DVB-S, DVB-S2, MPEG 2, MPEG 1, Transport stream, Program stream, DiSEqC, Skystar 2, audio, MP2, subtitles, teletext, transponder

Citace

Ivo Řezníček: Automatické ukládání digitálního satelitního vysílání, diplomová práce, Brno, FIT VUT v Brně, 2007

Automatické ukládání digitálního satelitního vysílání

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Ing. Igora Szökeho. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Ivo Řezníček
21. května 2007

Poděkování

Děkuji vedoucímu práce Ing. Igoru Szökemu za odbornou pomoc a všem, kteří se podíleli na výsledné podobě této práce.

© Ivo Řezníček, 2007.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1	Úvod	3
2	Systém DVB	5
2.1	DVB-T	6
2.2	DVB-S	6
2.3	DVB-S2	7
2.4	DVB-C	8
3	Standard MPEG2	9
3.1	Transport stream	11
3.2	Program stream	13
3.3	Paketizovaný elementární stream (PES)	14
3.4	Konverze mezi transport streamem a program streamem	15
3.5	Časovací model	15
3.6	Podmíněný přístup	15
3.7	Multiplexovací operace	15
3.8	Individuální operace prováděné nad elementárním datovým tokem	16
3.8.1	Demultiplexování	16
3.8.2	Synchronizace	16
3.8.3	Optimalizace vůči kompresní vrstvě	17
3.9	Systémové referenční dekodéry	17
4	MPEG2 transport stream	18
4.1	Syntaxe a sémantika transport stream paketu	19
4.2	Syntaxe a sémantika PES paketu	21
4.3	PSI tabulky	22
4.3.1	PAT tabulka	25
4.3.2	PMT tabulka	26
4.3.3	SDT tabulka	27
5	Hardwarové vybavení pro nahrávání dat	28
5.1	Výběr vhodného zařízení	28
5.2	DiSEqC sběrnice	30
5.3	Ovládání karty SkyStar2	31
6	Aplikace pro nahrávání satelitních dat	34
6.1	Aplikace pro správu dat	34
6.1.1	Modul pro práci s databází	37

6.1.2 Modul pro skenování satelitu	39
6.2 Plánovací soubor (schedule file)	41
6.3 Plánovač	43
6.4 Nahrávač dat	45
6.5 Skript pro následné zpracování	46
7 Závěr	47

Kapitola 1

Úvod

Hlavním požadavkem na vznik tohoto systému byla nutnost získávání velkého množství dat pro zdokonalování systémů pro zpracování řeči (zejména pro systémy identifikace jazyka). Možná řešení pro získávání dat jsou: internetová rádia, nahrávání a digitalizace rozhlasových a televizních stanic šířených pozemní cestou nebo nahrávání přímo digitálních dat. Srovnání všech těchto přístupů je naznačeno v tabulce 1.1.

	Inet. radio	DVB-T	DVB-C	DVB-S	Analog
Jazyky	cca 100	asi 3	asi 5	20-30	3-5
Kvalita	špatná	dobrá	dobrá	dobrá	špatná
Souběžné nahrávání	ano	ano	ano	ano	ne
Realizace	internet, sw	hw, sw	hw, sw	hw, parabola	anténa, tuner, ADC, sw
Celkově	použ.	ne	ne	použ.	ne

Tabulka 1.1: Přehled vlastností různých zdrojů dat

Internetová rádia jsou šířena často nejmenší možnou bitovou rychlostí, aby nebyly příliš zatěžovány servery, které je vysílají. To má za následek opravdu nekvalitní přenos záznamu, nejčastěji je používána komprese MP3, často též dochází ke snížení vzorkovací frekvence na 22050 či 16000 Hz. Je tu však výhoda, že na internetu je možné nahrávat mnoho stanic s řečovými nahrávkami, které je možné nahrávat paralelně. Paralelní nahrávání je ovšem omezeno kvalitou připojení k internetu (rychlostí příchozích dat).

DVB-T, DVB-C a DVB-S poskytuje velice kvalitní přenos řečových signálů, nejčastěji se používá formát MP2 s datovým tokem 192 Kbitů za sekundu. Norma umožňuje též použít formát AC3. Kabelová verze není v ČR ještě příliš rozšířená, terestriální verze se teprve buduje, satelitní vysílání je již v plném proudu a umožňuje paralelní příjem vysílaných stanic na jednom vysílacím kanálu. Prostředky na provozování těchto systémů jsou shrnuty v tabulce 1.1, pro satelitní vysílání je toto zařízení nejnáročnější.

Nahrávání analogového FM vysílání je mnohem náročnější, protože je nutné provádět analogově-číslíkovou konverzi se všemi operacemi s tím spojenými, tj. např. filtrování vstupního signálu (aby nedošlo k aliasingu). Tuto operaci jsme schopni provádět pouze jednu, jinak bychom potřebovali větší množství FM přijímačů, filtrů, AD převodníků atd.

Navíc v ČR není možné přijímat touto metodou mnohojazyčné vysílání, kromě češtiny možná navíc slovenštinu, polštinu a němčinu což není dostatečné.

Nejvhodnější řešení, které bylo vybráno, je získávat data, která jsou dostatečně kvalitní přímo ze satelitů a v digitální podobě (systém DVB). Zde je možné získat mnoho řečových signálů v různých jazycích (přibližně 20). Data mohou být nahrávána paralelně v rámci jednoho přenosového kanálu.

Tento dokument je dále dělen do kapitol, kde je popsán systém, ze kterého se budou řečová data získávat (kapitola 2). Datová vrstva toho systému MPEG2 (kapitola 3) a speciálně MPEG2 transport stream s rozšířeními, která jsou použita v systémech DVB (kapitola 4). Dále následuje kapitola 5, která se věnuje výběru vhodného hardwarového zařízení pro pořizování řečových záznamů a ovládání všech důležitých prvků, které se na tomto zařízení nachází. V poslední kapitole 6 je popsán návrh a realizace softwarové výbavy pomocí které bude možné masivně nahrávat řečová data ze satelitu.

Tato práce plynule navazuje na můj semestrální projekt *Automatizované ukládání digitálního satelitního vysílání*, který jsem řešil na Fakultě informačních technologií Vysokého učení technického v Brně.

Kapitola 2

System DVB

Při snaze digitalizovat stávající analogové vysílání je skupinou DVB Project od roku 1993 vyvíjen otevřený standard digitální televize **DVB** (**D**igital **V**ideo **B**roadcasting). Hlavní směry standardizace spočívaly v digitalizaci

- kabelové sítě,
- satelitního vysílání,
- pozemního vysílání.

Každá z výše uvedených soustav skrývá jiné problémy s přenosem obrazu a zvuku. Z toho plynou různé problémy na přenosové cestě. Společná je pro všechny přenosová média šířka přenosového kanálu, která se pohybuje v rozmezí 6-8MHz. Do této šíře pásma je nutné nějakým způsobem vměstnat digitální vysílání a zajistit spolehlivý provoz, který by obstojně odolal rušivým vlivům.

Nejmenší problémy s příjmem jsou zřejmě u přenosu kabelové televize, kdy poskytovatel je přímo spojen s přijímacím místem vodivým spojem. Mnohem horší to je se satelitním příjmem, kde je přenos realizován elektromagnetickým vlněním o vysokém kmitočtu (asi 10-12MHz), protože družice se nachází mimo atmosféru Země na geostaciární dráze. Signál tudíž musí atmosférou procházet, při tom může docházet k různým poruchám. Příjem je realizován fixně nastavenou parabolickou anténou, na níž je upevněn konvertor převádějící signál o vysokém kmitočtu na signál o kmitočtu přibližně řádově nižším (signál o kmitočtu 10,7 až 12,7 GHz je konvertován na signál o kmitočtu 0,9 až 1,7 GHz), tento signál je již možné klasickým koaxiálním kabelem přivést do tuneru přijímače.

Nejhorší situace nastává při pozemním příjmu a to zejména proto, že přijímání je nutné provádět nejen umístěnými anténami, ale též na pohyblivých místech (např. v autě), systém musí být tudíž robustní aby vyhověl požadavkům tohoto typu.

Skupina DVB Project tedy navrhla pro výše zmíněné použití tyto tři přenosové soustavy, jmenovitě

- **DVB-C** – digitalizace kabelové sítě,
- **DVB-S** a **DVB-S2** – satelitního vysílání,
- **DVB-T** – pozemního příjmu.

Všechny výše zmíněné DVB standardy mají jednu část společnou, celý DVB systém se dělí na vrstvu fyzickou a datovou. Každá verze DVB používá rozdílnou fyzickou vrstvu (z důvodu jiných technik přenosu) a datová vrstva je u vždy stejná, jedná se o rozšířený standard MPEG 2 transport stream. Rozšíření spočívá v multiplexu důležitých dodatkových informací do datového toku. Dodatkové informace udávají např. provozovatele sítě, názvy jednotlivých služeb (programů), informace pro dešifrování služeb s podmíněným přístupem, aktuální datum a čas a mnoho dalších.

Datové vrstvě (tedy systému MPEG 2 transport stream) se podrobně věnuje kapitola [3](#) a [4](#). V další části výkladu stručně vysvětlím základní principy přenosu na fyzických vrstvách všech důležitých DVB systémů, se zaměřením na satelitní vysílání tedy DVB-S a DVB-S2. Fyzická vrstva zajišťuje přenos dat mezi vysílačem a příjemcem. Jak již bylo zmíněno výše u každé z přenosových soustav mohou během přenosu nastávat odlišné druhy poruch působící na přenášený signál. Každá z uvedených soustav tudíž používá jiných vzájemně neslučitelných technik pro přenos, např. jiný druh modulace, v praxi to znamená že přijímací tuner a podpůrné obvody musí být pro každou soustavu jiné.

2.1 DVB-T

Přenosová soustava DVB-T byla ratifikována v roce 1997. Při vysílání ve standardu DVB-T musíme nejprve vytvořit MPEG 2 transport stream se všemi náležitostmi obsahující všechny vysílané programy. Následuje postup, který musí být proveden pro zakódování těchto dat do pozemního analogového kanálu.

- Adaptace a přeskládání multiplexovaného datového toku pro lepší rozložení výsledného spektra energie,
- vnější kódování (používá se kódování Reed-Solomon),
- aplikace vnějšího prokládání,
- vnitřní kódování (punctured convolutional code),
- vnitřní prokládání,
- mapování a modulace,
- OFDM vysílání.

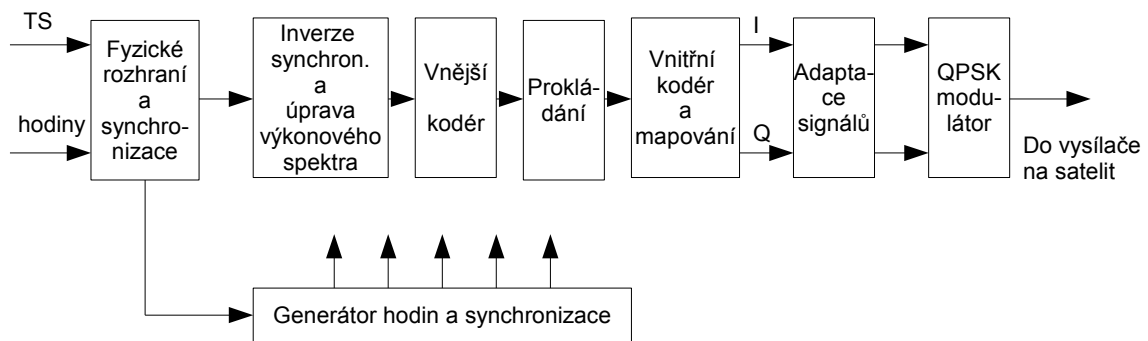
Podrobnější informace o tomto procesu je možné najít v [\[3\]](#).

2.2 DVB-S

Přenosová soustava DVB-S byla ratifikována v roce 1993. Při vysílání ve standardu DVB-S musíme nejprve vytvořit MPEG 2 transport stream se všemi náležitostmi, který obsahuje všechny vysílané programy. Následuje postup, který musí být proveden pro zakódování těchto dat do analogového kanálu, kterým se dříve vysílalo. Schéma tohoto postupu je znázorněno na obrázku [2.1](#). Podrobnější informace o tomto procesu je možné najít v [\[4\]](#).

- Adaptace transportního datového toku, oddělení signálu pro řízení časování celého systému,

- Inverze synchronizace a úprava výkonového spektra,
- Vnější kódování,
- Aplikace vnějšího prokládání,
- Vnitřní kódování (punctured convolutional code) a mapování,
- Adaptace signálů,
- QPSK modulace,



Obrázek 2.1: Princip kódování jednoho DVB-S kanálu

2.3 DVB-S2

Přenosová soustava DVB-S2 byla ratifikována v březnu 2005. Systém DVB-S2 je navržen tak, že využívá postupu technologie aby vyhověl náročným požadavkům dneška na satelitní vysílání. Je navržen tak robustně, že se předpokládá, že nebude nutné v dostatečně dlouhém časovém horizontu vytvářet nový satelitní přenosový systém. DVB-S2 přináší následující výhody:

- 30 % zvýšení efektivity než DVB-S.
- Zvýšení možnosti aplikací díky kombinaci funkcionality systému DVB-S (pro direct-to-home aplikace) and DVB-DSNG (pro profesionální aplikace).
- Přináší techniky, jako například adaptivní kódování, pro maximalizaci použití přenosového kanálu.

Technické charakteristiky systému DVB-S2 spočívají v rozšíření možnosti modulace, je možné použít následující systémy modulace: QPSK, 8PSK, 16APSK a 32APSK. Dále je též vylepšena dopředná oprava chyb. Z aplikačního hlediska systém DVB-S2 umožňuje přenášet data nejen ve formátu MPEG 2 transport stream ale je možné použít i jiný formát. V současné době je systém již používán např. na družicích Astra 1 a Hotbird 1 pro šíření HDTV vysílání v přenosové soustavě MPEG 4 AVC.

2.4 DVB-C

Přenosová soustava DVB-C byla ratifikována v roce 1993. Při vysílání ve standardu DVB-C musíme nejprve vytvořit MPEG 2 transport stream se všemi náležitostmi, který obsahuje všechny vysílané programy. Následuje postup, který musí být proveden pro zakódování těchto dat do kabelového analogového kanálu, kterým se dříve vysílalo v kabelové síti.

- Inverze synchronizace a přeskládání multiplexovaného datového toku,
- Reed-Solomon kódování,
- aplikace prokládání,
- konverze jednoho bajtu na symbol
- diferenční kódování,
- QAM modulace a úprava výsledku do výstupního fyzického rozhraní.

Podrobnější informace o tomto procesu je možné najít v [5].

Kapitola 3

Standard MPEG2

Systémy MPEG 2 spočívají v kombinování jednoho či více elementárních datových toků (elementary streams) videa, audia a jiných datových toků do jednoho či více toků, které jsou v podobě vyhovující pro uložení na nějaké paměťové médium nebo pro vysílání. Tyto systémy dále definují syntaktické a sémantické pravidla, které jsou předepsány v normě [7], a poskytují informace o synchronizaci pro výstupní dekodér, i při zhoršených příjmových podmínkách.

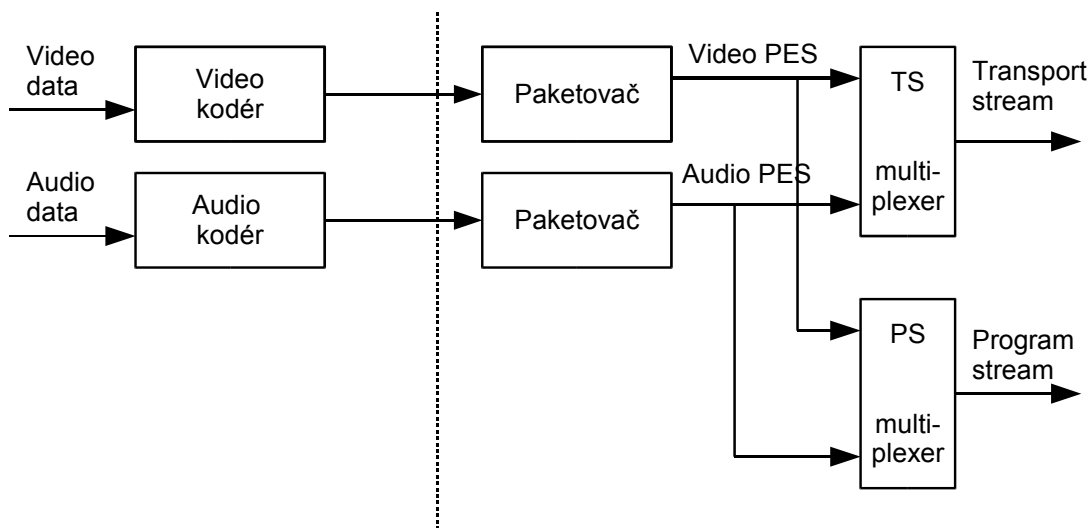
Systémové kódování je specifikováno ve dvou formách

- **Transport stream** (transportní datový tok)
- **Program stream** (programový datový tok)

Každý z nich je optimalizován pro odlišnou množinu aplikací. Oba jak transport stream tak i program stream obsahují informace, které jsou nezbytné pro synchronizaci dekódování a korektní přehrávání audio a video datových toků, přičemž je zabezpečeno, že nebude docházet k přetečení či podtečení dekódovacích vyrovnávacích pamětí. Synchronizační informace jsou ukládány v podobě dvou časových razítek. První časová razítka jsou použita pro synchronizované dekódování audio a video datových toků, druhá zajišťují synchronizaci celého datového toku jako celku. Transport i program stream je paketově orientovaný multiplex dat.

Základní přiblížení multiplexu jednoho video a jednoho audio datového toku je znázorněno na obrázku 3.1. Video a audio data jsou kódovaná, více o kódovacích procesech je možné zjistit v [8] a [2]. Jednotlivé kodéry produkují datový tok, který se označuje jako elementární datový tok (Elementary Stream – ES). Tento vstupuje do paketovače, který vytvoří PES pakety. Při tvorbě PES paketů mohou být přidávány informace pro použití PES paketů nezávisle na transport streamu či program streamu, při následném použití jednoho z těchto streamů není nutné tyto informace generovat.

Program stream spočívá v kombinování jednoho či více PES datových toků, které mají společný časový základ, do jednoho datového toku. Pokud konkrétní aplikace vyžaduje nemultiplexované elementární streamy uložené v separovaných streamech můžeme použít 2 program streamy, kde každý obsahuje jeden elementární datový tok, se společným časovým základem. V tomto případě hodnoty uložené v hlavičce obou program streamů v poli SCR by měly být konzistentní. Všechny elementární datové toky je možné dekódovat se synchronizací.



Obrázek 3.1: zjednodušené schéma multiplexu audio a video dat

Program stream je navržen pro použití v relativně bezporuchovém prostředí a je vhodný pro aplikace které mohou vyžadovat softwarové zpracování systémových informací jako například interaktivní multimediální aplikace. Pakety program streamu můžou mít proměnlivou ale dostatečně velkou délku.

Transport stream kombinuje jeden či více programů s různým časovým základem do jednoho celistvého datového toku. PES pakety vytvořené z elementárních streamů jednoho programu sdílejí stejný časový základ. Transport stream je navržen pro použití v prostředí, kde často dochází k chybám a poruchám, jako například přenos dat ztrátovými médii, kde může navíc docházet k tvorbě šumu. Pakety transport streamu mají fixní velikost 188 bajtů.

Program a transport stream jsou navrženy pro různé aplikace a jejich definice striktně nespĺňuje vrstvý model. Je možné a často k tomu dochází, konvertovat jeden do druhého, neznamená to ale že jeden je podmnožinou druhého. Získání obsahu jednoho programu z transport streamu a následná konverze do program streamu je možná díky společnému formátu PES paketů. To ovšem neznamená že všechny prvky program streamu jsou v transport streamu obsaženy, tzn. musí být synteticky vytvořeny, abychom zkonstruovali validní program stream. Transport stream překrývá více vrstev ve vrstvý model přenosu dat a je navržen pro efektivní a snadnou implementaci v aplikacích, které používají vysokých šířek pásma.

Rozsah syntaktických a sémantických pravidel se dále u obou specifikací systémů liší, syntaktická pravidla se týkají pouze vrstvy systémů, v žádném případě nezasahují do kompresní vrstvy, neboli kódování video a audio datových toků, naopak sémantická pravidla se týkají výhradně multiplexovaného datového toku jako celku.

3.1 Transport stream

Transport stream je přizpůsoben pro komunikační účely nebo pro ukládání jednoho či více programů kódovaných ve formátu MPEG 2 a jiných dat v prostředích, kde může docházet k chybám. Mluvíme o chybách typu bitové chyby v paketech či ztráty paketů.

Transport stream může mít konstantní či variabilní datový tok. V každém z těchto případů jednotlivé elementární streamy mohou mít též variabilní nebo konstantní datový tok. Syntaktické a sémantické omezení jsou stejné v každém z těchto případů. Rychlost transportního datového toku je definována hodnotami a časovým umístěním pole *PCR* (Program clock reference – programové hodiny), každý program má vlastní hodiny.

Transport stream může být vytvořen jakoukoliv metodou, která má za výsledek validní datový tok. Je možné ho vytvořit například z elementárních datových toků, z program streamu nebo z jiného transportního streamu, který může obsahovat jeden či více datových toků.

Transport stream je navržen tak aby nad ním bylo možné s minimálním úsilím vykonávat několik důležitých operací. Tyto operace jsou:

1. Příjem kódovaných dat jednoho programu, dekodování a presentace výsledku jak je naznačeno na obrázku 3.2.
2. Extrakce paketů jednoho programu z transport streamu a vytvoření výstupu, který bude obsahovat pouze jeden program jak je naznačeno na obrázku 3.3
3. Extrakce paketů jednoho či více programů z jednoho či více transportních datových toků a tvorba jiného transport streamu.
4. Konverze jednoho programu v transport streamu do program streamu jak je vyobrazeno na obrázku 3.4.
5. Konverze z program streamu do transport streamu, dále přenos přes ztrátový kanál či ztrátové prostředí a poté znovuobnovení často identického program streamu.

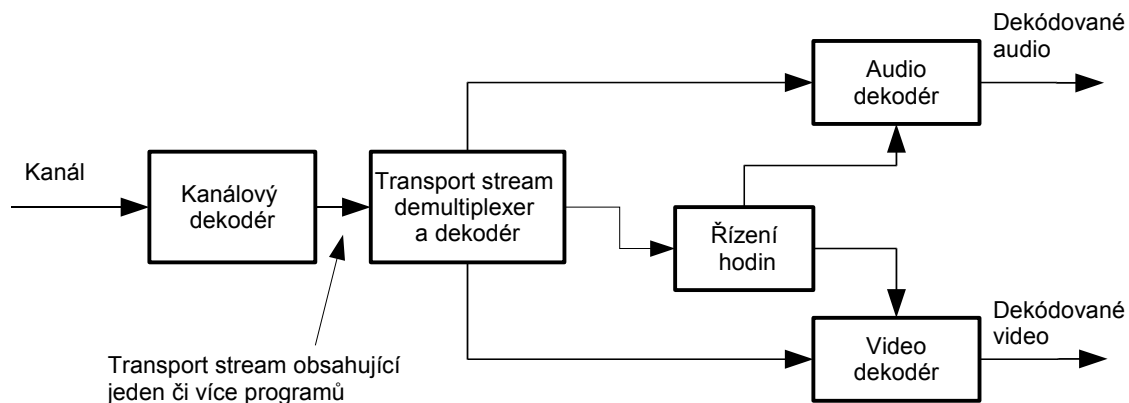
Obrázek 3.2 a 3.3 zobrazuje prototyp demultiplexujícího a dekodovacího systému, jehož vstupem je transport stream. Obrázek 3.2 demonstruje první případ použití, kdy je transport stream přímo demultiplexován a dekodován. Transport stream je navržen a zkonstruován ve dvou vrstvách

1. systémová vrstva,
2. kompresní vrstva.

Se systémovou vrstvou, která “obaluje” kompresní vrstvou, pracuje transport stream dekodér, odděluje ji a dále posílá pouze kompresní vrstvu ke zpracování do audio a video dekodérů.

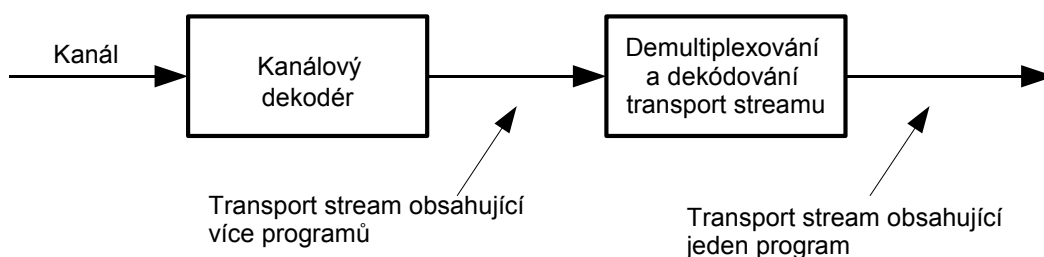
Operace, které provádí prototyp dekodéru jsou aplikovány buď na celý transportní tok (“multiplexovací operace”), nebo na každý elementární stream (“streamové operace”). Transport stream je rozdělen na dvě vrstvy, jedna pro multiplexovací operace (vrstva transport stream paketů), a druhá pro streamové operace (vrstva PES paketů).

Prototyp transport stream dekodéru, který též obsahuje audio a video dekodér je kvůli demonstraci funkce zobrazen na obrázku 3.2. Architektura není ideální – některé funkce systémového dekodéru, jako například časovací prvek, by mohly být distribuovány společně s elementárními streamy. Stejně tak indikace chyb, které by byly detekovány kanálovým dekodérem a jejich přenos do audio a video dekodérů může být provedeno mnoha způsoby. Komunikační kanály nejsou v diagramu zobrazeny. Pomocí transport streamu je možné též přenášet jiné než audio či video data, další zpracování těchto dat není na diagramu zobrazeno.



Obrázek 3.2: Prototyp dekodéru transport streamu

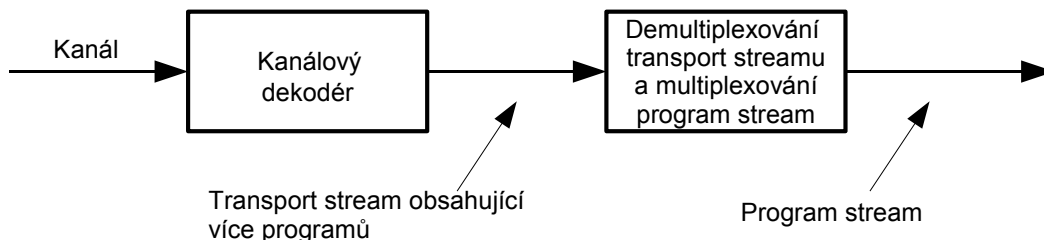
Obrázek 3.3 znázorňuje druhý případ, kde transport stream obsahuje více programů a je transformován do transport streamu s jedním programem. V tomto případě re-multiplexovací operace může vyžadovat korekci programových hodin (PCR) z důvodu změny pozice PCR hodnot v bitovém toku.



Obrázek 3.3: Prototyp transport stream multiplexování

Obrázek 3.4 demonstruje případ kdy víceprogramový transport stream je nejdříve demultiplexován a posléze konvertován do program streamu.

Na obrázku 3.3 a 3.4 je zobrazeno, že je možné a často se provádí konverze mezi různými typy transport streamu. Existují záznamy v transportním i program streamu, které umožňují zobrazené konverze. Neexistují žádné speciální požadavky na implementaci demultiplexoru a dekodéru, které zahrnují všechny tyto funkce.



Obrázek 3.4: Prototyp konverze z transport streamu do program streamu

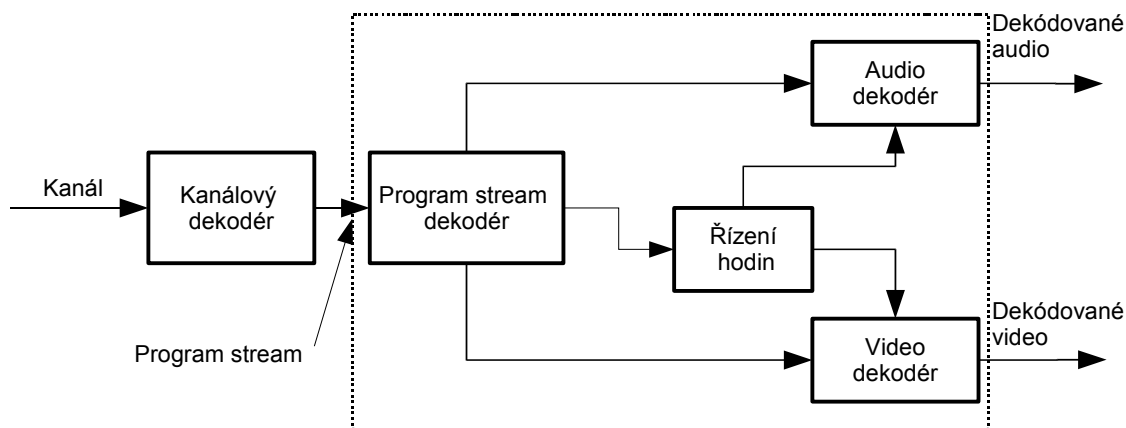
3.2 Program stream

Program stream je koncipován pro uložení jednoho programu, který obsahuje audio, video a popřípadě jiná kódovaná data (např. teletext), v prostředích, kde nejčastěji nedochází k chybám a pokud nastanou jsou systémově zpracovávány. Program stream může být proměnlivé či konstantní datové rychlosti, stejně jako jeho jednotlivé elementární streamy. Syntaktické i sémantické omezení jsou v obou případech stejné.

Prototyp audio/video program stream dekodéru je zobrazen na obrázku 3.5. Architektura není ideální – některé funkce systémového dekodéru, jako například časovací prvek, by mohly být distribuovány společně s elementárními streamy. Komunikační kanály nejsou v diagramu zobrazeny. Pomocí Transport streamu je možné též přenášet jiné než audio či video data, další zpracování těchto dat není na diagramu zobrazeno.

Prototyp program stream dekodéru je složen ze systémového, audio a video dekodéru. Pokud je v jednom kanálu uloženo více audio či video streamů musí kanálový dekodér odeslat dále ke zpracování jen jeden program v podobě program streamu, MPEG2 standard nijak nestanovuje kanálové uložení dat.

Do dekodéru vstupuje program stream z něhož jsou extrahovány informace potřebné k časování datových toků. Poté dochází k demultiplexu elementárních audio a video datových toků, které vstupují do jednotlivých dekodérů. Extrahované časovací informace slouží k synchronizaci audio a video datových toků, které jsou v příslušných dekodérech dekodovány.



Obrázek 3.5: Prototyp program stream dekodéru

Program stream je navržen ve dvou vrstvách.

1. systémová vrstva,
2. kompresní vrstva.

Program stream jako takový obsahuje systémovou vrstvu, která obaluje kompresní vrstvu, v program stream dekodéru je systémová vrstva odstraněna a do audio a video dekodérů vstupují pouze data na kompresní vrstvě.

Operace dekodéru se dají rozdělit do dvou skupin, multiplexovací a streamově orientované. Prvně zmíněné operace se provádí program stream dekodér, jsou to tedy operace prováděné na systémové vrstvě, druhé se provádějí nad PES pakety, které tvoří kompresní vrstvu.

3.3 Paketizovaný elementární stream (PES)

Transport stream i program stream je konstruován z PES paketů, jak již bylo zmíněno. PES pakety mohou být v některých případech beze změny použity při konverzi mezi transport a program streamem. Počet bajtů PES paketu může být mnohem větší než je velikost jednoho paketu transport streamu. Plynulá sekvence PES paketů jednoho elementárního streamu, která je identifikována stejným ID, může být použita k vytvoření PES streamu.

PES stream je logická struktura, která je užitečná při obecném popisu funkce komponent při zpracování program streamu či transport streamu, avšak není navržen k účelům přenášení audio či video dat mezi dvěma komponentami. Pokud aplikace vyžaduje datový tok, obsahující pouze jeden elementární stream, může bez problémů používat program stream či transport stream s jedním elementárním datovým tokem. Takový stream obsahuje všechny potřebné systémové informace. Několikanásobný transport či program stream, každý obsahující jeden elementární stream, může být vytvořen se společným časovým základem, tím pádem mohou nést kompletní program, například audio, video a teletext.

3.4 Konverze mezi transport streamem a program streamem

Konverze mezi transport streamem a program streamem je možná díky existenci PES paketů. PES pakety mohou být, při dodržení určitých pravidel, bitově okopírovány z jednoho datového toku do druhého. K tomu je nutné identifikovat správné pořadí jednotlivých PES paketů, všechny PES pakety musí obsahovat počítadlo paketů.

3.5 Časovací model

Systémy, video a audio zpracování, všechny prvky přenosového řetězce mají časovací model, kdy zpoždění od začátku do konce celé posloupnosti operací, tedy od vstupních kodérů až po výstupní dekodéry, je konstantní. Toto zpoždění je dáno součtem zpoždění výpočtu kodéru, vyrovnávacích operací kodéru, multiplexování, přenosu či ukládání, následného demultiplexování, vyrovnávacích operací dekodéru, dekódování a výsledné prezentace obsahu. Časovací model určuje, že všechny video snímky a všechny audio vzorky budou prezentovány pouze jednou a bude zajištěno že interval mezi snímky a audio vzorky bude stejný na straně dekodéru i na straně kodéru. Kódování systémových datových toků je navrženo tak, že obsahuje časovací informace, které je možné použít k implementaci systému se stejným časovým zpožděním. Prakticky není možné sestavit systém, který bude mít přesné časové zpoždění ze vstupu na výstup, korekce tohoto stavu musí provádět dekodér. Časování musí být dodržováno při tvorbě všech validních datových toků.

Veškeré časování je specifikováno v podobě značek (razítek) společných systémových hodin, které jsou odvozeny z aktuálního času. V program streamu mohou být tyto hodiny vyjádřeny poměrem k časování jednotlivých vzorků audio či video dat, nebo mohou mít úplně jinou frekvenci, přičemž je stále zajištěno jednotné celkové zpoždění celé přenosové cesty. Transport stream má hodiny vždy omezeny na určitý poměr vůči hodinám audio a video dat, kvůli nutnosti snadnějšího obnovení hodin v dekodéru.

3.6 Podmíněný přístup

Šifrování a kódování pro podmíněný přístup k programu uloženém v program či transport streamu je specifikován v definicích datových toků. Transport stream umožňuje šifrování přímo na úrovni systémové vrstvy, celý paket je tedy možné zašifrovat tak že není možné rozeznat ani hlavičky PES paketů. PES pakety samy o sobě podporují šifrování obsažených dat, dále je možné ukládání privátních dat. Na vrstvě PES paketů je tudíž možné šifrovat i program stream. Tohoto principu je využíváno na video nosičích DVD, kdy u některých PES paketů je nastaven příznak zašifrování, celý PES paket je tedy nutné před odesláním do dekodéru elementárního datového toku navíc dešifrovat.

3.7 Multiplexovací operace

Multiplexovací operace zahrnují koordinaci příjmu dat kanálu, úpravu hodin a správu vyrovnávacích pamětí. Tyto úkoly spolu úzce souvisí. Pokud je možné kontrolovat rychlost dat v kanálu, tak může být docíleno toho, že nedochází k podtečení ani přetečení vyrovnávacích pamětí. V opačném případě musí dekodéry jednotlivých elementárních datových toků zajišťovat časování dat příchozích z kanálu aby se předešlo podtečení či přetečení vyrovnávacích pamětí.

Program stream obsahuje pakety, které v hlavičce obsahují informace k výraznému ulehčení těchto operací. Hlavička specifikuje závazná časová razítka ve kterých každý bajt datového toku vstupuje z kanálu do dekodéru program stream dekodéru. Tyto časové informace slouží jako reference pro korekci hodin a pro správu vyrovnávacích pamětí. Obdobně transport stream je složen z paketů, kde hlavička obsahuje časy, v kterých každý bajt vstupuje z kanálu do transport stream dekodéru. Jejich použití má přesně stejné použití jako v případě program streamu.

Dodatečné multiplexovací operace spočívají ve schopnosti dekodéru vybrat, které prostředky jsou nutné pro dekódování program či transport streamu. První paket program streamu obsahuje právě tyto informace. Např. maximální rychlost celého datového toku, či maximální počet video datových toků. Transport stream obsahuje podobné globální informace.

3.8 Individuální operace prováděné nad elementárním datovým tokem

Do těchto operací se zařazuje demultiplexování a synchronizace přehrávání 2 a více elementárních datových toků.

3.8.1 Demultiplexování

Program stream je vytvářen pomocí multiplexu elementárních datových toků, transport stream je tvořen pomocí multiplexu elementárních datových toků nebo z jiných program streamů nebo z jiných transport streamů. Elementární datové toky mohou obsahovat privátní, vyhrazená či vyplňovací datové toky, jako dodatek k audio či video datovému toku. Dále jsou tyto datové toky dočasně rozděleny do paketů a tyto pakety jsou serializovány. PES pakety obsahují kódovaná data výhradně jednoho elementárního datového toku.

V program streamu jsou povoleny různé délky jednotlivých paketů, existují však určitá omezení, více o těchto omezeních je možné zjistit v [7]. Pro pakety transport streamu je délka jednoznačně stanovená na 188 bajtů. Velikosti PES paketů jsou též proměnné, velikost závisí na konkrétní aplikaci.

3.8.2 Synchronizace

Synchronizace mezi jednotlivými elementárními datovými toky je prováděna pomocí takzvaných prezentačních časových razítek (Presentation Time Stamps – PTS), které jsou obsaženy v program streamu i transport streamu. Časová razítka jsou generována s kmitočtem 90 kHz, avšak systémové hodiny (System Clock Reference – SCR), programové hodiny (Program Clock Reference – PCR) a volitelné hodiny jednotlivých elementárních datových toků (Elementary Stream Clock Reference – ESCR) mají zvětšené rozlišení a jsou generovány s kmitočtem 27 MHz. Dekódování N elementárních datových toků je synchronizováno přizpůsobením dekodování všech datových toků vůči hlavní společné časové základně, než korekcí dekodování jednoho datového toku k časové základně jiného. Hlavní společná časová základna může být zvolena jako hodiny jednoho z N dekodérů, hodiny celého datového toku či externí hodiny.

Každý program obsažený v jednom transport streamu může mít svou vlastní časovou základnu. Časové základny různých programů v transport streamu mohou být různé.

Jelikož prezentační časová razítka se striktně používají k dekodování jednotlivých elementárních datových toků, jsou uloženy v PES paketech v program streamu i transport streamu. Synchronizace ze začátku celého přenosového řetězce na konec nastává, když kodér ukládá časové razítka ve chvíli snímání vstupních dat, tyto jsou propagovány pomocí datového toku k dekodéru a ten tyto časová razítka používá k přesnému plánování prezentace výstupu.

3.8.3 Optimalizace vůči kompresní vrstvě

Vrstva PES paketů je v několika ohledech nezávislá na kompresní vrstvě, ale ne ve všech. Nezávislá je v tom, že při paketování elementárních datových toků nebere žádné ohledy na hlavičky těchto datových toků. Například hlavičky video elementárního datového toku se mohou vyskytovat kdekoli v PES paketu a můžou tedy být rozděleny do dvou PES paketů. Ale prezentační časová razítka, která jsou uložena v hlavičce PES paketu, jsou spojena s prezentací dat, která jsou kódována na kompresní vrstvě. Pokud elementární datový tok vyhovuje normě H.262 [8], velikosti PES paketů by měly být zarovnaný tak, že k výše uvedeným anomáliím nedochází.

3.9 Systémové referenční dekodéry

Norma ISO-IEC 13818 používá tzv. systémový referenční dekodér (System Target Decoder - STD), jeden pro transport stream (T-STD) a jeden pro program stream (P-STD), k vytvoření formalismu pro vztahy mezi časováním a vyrovnávacími paměťmi. Jelikož je systémový dekodér parametrizován pomocí polí (definici je možné najít v [7]), např. velikost vyrovnávací paměti, každý elementární datový tok vyžaduje jiné parametrizování tohoto modelu. Kodéry by měly produkovat datové toky, které vyhovují specifickým omezením. Fyzické dekodéry mohou předpokládat že datový tok, který je dodáván je přehratelný na modelovém referenčním dekodéru

Kapitola 4

MPEG2 transport stream

Transport stream umožňuje kombinování jednoho či více programů do jednoho datového toku. Data každého elementárního datového toku musí být též multiplexována a doplněna informacemi, díky kterým je možné synchronizovanou prezentaci audio a video datových toků.

Elementární streamy jsou přenášeny pomocí PES paketů. Ten se skládá z hlavičky a dat, které přenáší. PES pakety jsou vloženy do transport stream paketu a to tak, že každý PES paket je vždy umístěn na začátku transport stream paketu.

Hlavička PES paketu začíná 32-bitovým startovacím kódem, který identifikuje stream nebo typ streamu, do kterého tento paket náleží. Obsahem může též být dekodovací a prezentační časové razítko (DTS, PTS) a jiné volitelné informace. Datový prostor obsahuje variabilní počet bajtů elementárního datového toku, bez případných překryvů.

Pakety transport streamu obsahují 4-bajtovou hlavičku, která obsahuje 13-bitovou identifikaci paketu (PID). S multiplexovanými audio a video datovými streamy je nutné dále přenášet metainformace o vysílaných programech, tyto jsou organizovány do tabulek (PSI tabulek). Pakety, které mají stejný PID identifikátor přenáší pouze jeden elementární datový tok.

PSI tabulky definovány v transport streamu:

- **PAT** (Program Association Table),
- **PMT** (Program Map Table),
- **CAT** (Conditional Access Table),
- **NIT** (Network Information Table).

PSI tabulky, které přináší norma systému DVB (detailní informace v [6]):

- **SDT** (Service Description Table),
- **TDT** (Time and Date Table),
- **EIT** (Event Information Table),
- **RST** (Running Status Table).

- TOT (Time Offset Table).
- BAT (Bouquet Association Table).
- ST (Stuffing Table).

Pro účely přijímání řečových dat bude potřebné získávat data z tabulek: PAT, PMT, a SDT.

4.1 Syntaxe a sémantika transport stream paketu

Syntaxe jednoho transport stream paketu je znázorněna v tabulce 4.1. Sémantika jednotlivých polí je popsána níže.

Název pole	počet bitů
sync_byte	8
transport_error_indicator	1
payload_unit_start_indicator	1
transport_priority	1
PID	13
transport_scrambling_control	2
adaptation_field_control	2
continuity_counter	4
adaptation_field	184*8 - N*8
data_byte	N*8

Pozn. – N je rovno počtu bajtů, které jsou paketem přenášeny.

Tabulka 4.1: Syntax transport stream paketu

sync_byte – synchronizační bajt, hodnota je nastavena na '0100 0111' (0x47 hexadecimálně).

transport_error_indicator – jednobitový příznak. Pokud je nastaven na '1' indikuje jednu neopravitelnou bitovou chybu v celém transport stream paketu. Tento bit by měl být nastaven zařízeními pracující mimo vrstvu transport streamu. Když je jednou nastaven neměl by být již nikdy resetován a nastaven do '0' pokud bitová chyba nebyla opravena.

payload_unit_start_indicator – jednobitový příznak, který má zásadní rozlišovací význam, zda transport stream paket přenáší PES pakety nebo PSI tabulky z pohledu fyzického začátku těchto paketů.

Pokud transport stream přenáší PES pakety je tento příznak následující význam: hodnota '1' znamená, že první bajt přenášený v transport stream paketu je taktéž prvním bajtem PES paketu, '0' znamená že žádný PES paket nezačíná.

Pokud transport stream přenáší PSI tabulky má tento příznak následující význam: hodnota '1' znamená, že první bajt přenášený v transport stream paketu je ukazatel na prvním bajt PSI tabulky, v opačném případě je tento příznak nastaven na '0'.

Pro nulové pakety je by měl být tento příznak nastaven na '0'. Pokud transport stream přenáší privátní data není nastavení tohoto příznaku normou definováno.

transport_priority – jednobitový příznak. Pokud je nastaven na '1' znamená to, že tento paket má větší prioritu než jakýkoliv jiný s nastaveným tímto příznakem na '0'. Tento příznak může být měněn kanálovým kóděrem či dekodérem transportního datového toku.

PID – třináctibitový identifikátor, který popisuje typ dat, který je přenášen v paketu. Celkový souhrn významů jednotlivých PIDů je uveden v tabulce 4.2.

Hodnota (hex- adecimálně)	Popis
0x0000	PAT tabulka
0x0001	CAT tabulka
0x0002	TSD tabulka
0x0003 – 0x000F	Vyhrazeno
0x0010 – 0x1FFE	Tyto hodnoty mohou být použity jako síťový PID, PMT PID, elementární PID nebo pro jiné účely
0x1FFF	Nulový paket

Pozn. – Transport pakety s PIDem 0x0000, 0x0001 a 0x0010 – 0x1FFE je možné použít k přenášení programových hodin (PCR).

Tabulka 4.2: Syntax transport stream paketu

transport_scrambling_control – Tato dvojice bitů specifikuje zda jsou data přenášená v paketu šifrována. Hlavička paketu a případné pole pro přizpůsobení (adaptation field) nesmí být šifrováno. V nulovém paketu musí být toto pole vždy nastaveno na '00'. Následující tabulka zobrazuje použití šifrování v systému DVB, kde se šifrování provádí pomocí dvou klíčů, které se cyklicky opakují (Even Key a Odd Key), vše je prezentováno v tabulce 4.3.

Hodnota	Popis
00	Nešifrováno
01	Šifrováno pomocí tzv. Even Key
10	Šifrováno pomocí tzv. Odd Key
11 – 0x000F	Vyhrazeno

Tabulka 4.3: Schéma šifrování obsahu v systému DVB

adaptation_field_control – Tyto dva bity specifikují čím je následována hlavička paketu transport streamu. Je to buď pole pro přizpůsobení nebo přímo přenášený obsah nebo obě datové struktury za sebou, vše je specifikováno v tabulce 4.4.

continuity_counter – Čtyřbitové pole, které je inkrementováno s každým paketem transport streamu se stejným PIDem. Při inkrementaci dochází k přetečení tohoto čítače, hodnota se vždy vrací do 0. Tento čítač se neinkrementuje pokud je *adaptation_field_control* roven '00' nebo '10'.

Hodnota	Popis
00	Vyhrazeno pro budoucí použití
01	Paket obsahuje pouze přenášená data
10	Paket obsahuje pouze pole pro přizpůsobení
11	Paket obsahuje pole pro přizpůsobení, které je následováno přenášenými daty

Tabulka 4.4: Módy obsahu transport stream paketu

V transport streamech je možné provádět duplikaci paketů jako sekvence dvou posobějdoucích paketů se stejným PIDem. Tyto dva pakety mají stejnou hodnotu v poli *continuity_counter*, přičemž *adaptation_field_control* musí být roven '00' nebo '10'.

Hodnota tohoto pole je neustále zvyšována v závislosti na nastavení pole *continuity_counter* předchozího transport stream paketu se stejným PIDem a v závislosti na nastavení pole *adaptation_field_control*. V případě nulového paketu není nastavení tohoto pole definováno.

data_byte – Obsahuje posloupnost bajtů PES paketu, PSI tabulek, zarovnávací bajty, které následují po PSI tabulkách nebo privátní bajty. Velikost těchto dat je závislá na zbytku prostoru v paketu, tedy standardně 184 bajtů, pokud paket navíc obsahuje pole pro přizpůsobení je délka snížena o velikost tohoto pole.

adaptation_field – Poměrně rozsáhlé datové pole, které umožňuje přenášet velké množství informací pro synchronizaci přenosové cesty mezi kodérem a dekodérem transport stream datového toku a jednotlivými kodéry a dekodéry elementárních datových toků. Často se též používá k výplni části paketu transport streamu, např. při ukládání poslední části PES paketu, kdy není využit celý prostor paketu. V tomto případě pole obsahuje výplňové bajty a jeho délka pole nastavena přesně tak aby součet délky tohoto pole a velikosti zbytku PES paketu byl roven 184.

Detailní informace o přesné syntaxi a sémantice všech datových prvků, které se v tomto poli mohou nacházet je možné získat v [7].

4.2 Syntaxe a sémantika PES paketu

PES pakety jsou, jak již bylo zmíněno dříve, nosné médium pro přenášení všech typů elementárních datových toků. V této sekci zevrubně popíšu některé doplňkové informace, které mohou být v záhlaví PES paketu přenášeny. Hodně zjednodušená syntax PES paketu je zobrazena v tabulce 4.5.

Název pole	Počet bitů
packet_start_code_prefix	24
stream_id	8
PES_packet_length	16
'01'	2
PES_scrambling_control	2
PES_priority	1
data_alignment_indicator	1
copyright	1
original_or_copy	1
PTS_DTS_flag	2
ESCR_flag	1
ES_rate_flag	1
DSM_trick_mode_flag	1
additional_copy_info_flag	1
PES_CRC_flag	1
PES_header_data_length	8
<i>additional header data</i>	
<i>PES stream data</i>	

Tabulka 4.5: Syntax PES paketu

Celý paket je možné rozdělit na tři základní části

- pevná hlavička,
- rozšířená hlavička,
- data přenášená v PES paketu.

Pevná hlavička je v paketu obsažena vždy, obsahuje identifikaci datového toku v poli *stream_id*. Hodnoty 0xE0 - 0xE7 specifikuje první až osmý video datový tok a 0xC0 - 0xC7 audio datové toky. První pole je vždy nastaveno na hodnotu 0x0001. Položka *PES_scrambling_control* se používá pro specifikaci toho, zda jsou přenášená data šifrována. Příznaky od *PTS_DTS_flag* po *PES_extension_flag* specifikují obsah rozšířené hlavičky. Pokud jsou tyto příznaky nastaveny obsahuje rozšířená hlavička postupně datová pole spřažená s nastavenými příznaky. Všechny tyto datové struktury jsou přesně definovány v [7]. Z pohledu nahrávání řečových dat je nejdůležitější struktura spřažená s *PTS_DTS_flag* příznakem, která obsahuje prezentační časová razítka. Díky těmto razítkům je možné synchronizovat elementární datové toky, např. audio a teletext. Velikost rozšířené hlavičky je stanovena v položce *PES_header_data_length*.

4.3 PSI tabulky

PSI tabulky (Program Specific Information) obsahují normovaná data a privátní data, která umožňují dekodérům demultiplexování programů. Programy jsou sestaveny z jednoho či více elementárních datových toků, přičemž každý je možné rozlišit pomocí PIDu. Programy,

elementární datové toky či jejich části mohou být šifrovány pro podmíněný přístup. Avšak informace uložené v těchto tabulkách se nešifrují.

V transportních streamech jsou tyto informace klasifikovány do pěti základních tabulek (znázorněno v tabulce 4.6). Některé tabulky mají přesně specifikovaný PID, u některých je PID volitelný. Tyto tabulky jsou uloženy do transport stream paketu a jsou multiplexovány do transport streamu. V normě [7] je specifikováno, že pomocí PSI tabulek je možné přenášet privátní tabulky, přičemž význam přenosu norma nestanovuje. Může se jednat např. o proprietární vysílání EPG informací (Electronic Program Guide – elektronický program stanic).

Název tabulky	PID	Popis
Program Association Table (Seznam programů)	0x0000	Asociuje číslo programu a PID PMT tabulky
Program Map Table	přiřazen v PAT	Specifikuje PID hodnoty jednotlivých komponent programu
Network Information Table	přiřazen v PAT	Specifikuje fyzické parametry sítě, jako např. FDM frekvence počet transpondérů atd.
Conditional Access Table (Seznam šifrovacích zařízení)	0x0001	Asociuje jeden či více EMM streamů s jeho unikátním PIDem
Service Description Table	0x0013	Parametry jednotlivých služeb

Tabulka 4.6: Členění základních PSI tabulek

Jednotlivé tabulky mohou mít variabilní délku začátek sekce je indikován pomocí osmi-bitové položky *pointer_field*, která určuje ukazatel na první bajt tabulky. Celá tabulka je umístěna v transport stream paketu, pokud paket navíc obsahuje pole pro přizpůsobení, je tabulka umístěna až za tímto polem.

V transport streamu se mohou nacházet výplňové bajty, tyto se používají pro vyplňování zbytku paketu pokud je délka tabulky nebo délka posledního segmentu tabulky menší než vyhrazený rozsah bajtů v transport stream paketu. V takovýchto případech je do paketu vložena celá tabulka a nevyužitá místa je zaplněna těmito výplňovými bajty, jejich hodnota je stanovena na 0xFF (hexadecimálně). Dekodéry těchto tabulek výplňové bajty zahazují.

Každý transport stream musí obsahovat PAT tabulku (Program Association Table), jejíž PID má hodnotu 0x0000. Tato tabulka obsahuje kompletní seznam všech programů, které jsou v transport streamu obsaženy.

Pokud je jeden či více elementárních datových toků šifrován, musí transport stream obsahovat CAT tabulku (Conditional Access Table), jejíž PID má hodnotu 0x0001, a která obsahuje deskriptory systémů podmíněného přístupu. Tyto systémy jsou asociovány s jednotlivými šifrovanými elementárními datovými toky.

Každý transport stream obsahuje jeden či více paketů s PID hodnotami, které jsou specifikovány v PID tabulce jako pakety obsahující mapovací sekci (Program Map Table). Každý program definovaný v PAT tabulce je popsán unikátní PMT tabulkou, která ho plně definuje. Pokud program obsahuje privátní data, které jsou přenášeny pomocí ele-

mentárního datového toku musí být jeho PID v PMT tabulce specifikován. Jiná privátní data mohou v transport streamu existovat separovaně od jakéhokoliv programu. Hodnota položky *table_id* v této tabulce je nastavena na hodnotu 0x02.

Network Information Table (NIT) je volitelná její obsah je privátní a obsahuje informace o síti či provozovateli, který vysílací kanál provozuje. Pokud je vysílána její PID má hodnotu, která je uživatelsky specifikovaná v tabulce PMT jako *network_PID*, viz tabulka 4.8. Specifikace se provádí nastavením hodnoty pole *program_number* na vyhrazenou hodnotu 0x0000 a nastavením hodnoty pole *network_PID* na požadovanou hodnotu. Tato hodnota může být v rozmezí 0x0010 – 0x1FFE. Přesná syntax a sémantika je uvedena v [6].

Maximální velikost PSI tabulky je stanovena na 1024 bajtů. Velikost Privátní sekce je stanovena max. na 4096 bajtů. Všechny tabulky mají společné 16-bitové pole *table_id*, které má přesně definované hodnoty pro jednotlivé typy tabulek. Specifikace pro nejdůležitější tabulky v systému DVB je uvedena v 4.7.

Hodnota	popis
0x00	PAT tabulka
0x01	CAT tabulka
0x02	PMT tabulka
0x03	TSDT tabulka
0x40	NIT tabulka, aktuální síť
0x41	NIT tabulka, jiná síť
0x42	SDT tabulka, aktuální síť
0x46	SDT tabulka, jiná síť
0x4E	EIT tabulka, aktuální síť
0x4F	EIT tabulka, jiná síť

Tabulka 4.7: Přiřazení hodnot položky *table_id*

V dalších odstavcích bude blíže uvedena syntaxe a sémantika všech tabulek, které jsou významné z hlediska budování systému pro masivní získávání řečových dat. Při popisu několikanásobného výskytu či podmíněného výskytu je z důvodu jednodušší prezentace využita pseudosyntaxe jazyka C. Přesnou definice ostatních tabulek je možné nalézt v příslušných normách [7] a [6].

4.3.1 PAT tabulka

Přesná syntaxe PAT tabulky je uvedena v tabulce 4.8. Pro nás nejdůležitější je zjištění relace mezi identifikátorem služby *program_number* a PIDem PMT tabulky, který je uveden v poli *program_map_PID*. Je tu též specifikován PID NIT tabulky, a to tak, že pole *program_number* je nastaveno na hodnotu 0x00.

Název datové položky	počet bitů
table_id	8
section_syntax_indicator	1
'0'	1
reserved	2
section_length	12
transport_stream_id	16
reserved	2
version_number	5
current_next_indicator	1
section_number	8
last_section_number	8
for (i = 0; i < N; i++) {	
program_number	16
reserved	16
program_map_PID	13
}	
CRC_32	32

Tabulka 4.8: Syntax PAT tabulky

4.3.2 PMT tabulka

PMT tabulka přináší mapování mezi číslem programu a elementárními datovými toky, které tento program tvoří. Jedná se o kompletní definici, která specifikuje podmnožinu transport streamu tvořící daný program. PID hodnoty této tabulky stanovuje kódér transportního datového toku. Každá služba (program) má v transport streamu svou PMT tabulku. Datová položka *program_number* specifikuje číslo služby, stejné, které je specifikováno v PAT tabulce. Důležitou informací je PCR PID, pomocí kterého je přenášeno časování celé služby. Tabulka dále umožňuje přenášet deskriptor ke každému PIDu a ke službě jako celku. Syntax celé tabulky je znázorněna v tabulce 4.9.

Název datové položky	počet bitů
table_id	8
section_syntax_indicator	1
'0'	1
reserved	2
section_length	12
program_number	16
reserved	2
version_number	5
current_next_indicator	1
section_number	8
last_section_number	8
reserved	3
PCR_PID	13
reserved	4
program_info_length	12
for (i = 0; i < N; i++) {	
descriptor()	
}	
for (i = 0; i < M; i++) {	
stream_type	8
reserved	3
elementary_PID	13
reserved	4
ES_info_length	12
for (i = 0; i < M; i++) {	
descriptor()	
}	
}	
CRC_32	32

Tabulka 4.9: Syntax PMT tabulky

4.3.3 SDT tabulka

Tabulka SDT (syntax v tab. 4.10) obsahuje informace o názvu a stavu jednotlivých služeb, jádrem celé tabulky je seznam umístěný u konce tabulky, kde je možné nalézt položku *service_id*, která je shodná s číslem služby v tabulce PAT. Dále obsahuje identifikaci zda je služba šifrovaná (*free_ca*), stav služby (*running_status*) a deskriptory služby. Deskriptory služby obsahují informaci o provozovateli služby a název služby. Podrobné informace o možných stavech služby a kompletní popis obsahu této tabulky lze nalézt v [6].

Název datové položky	počet bitů
table_id	8
section_syntax_indicator	1
reserved_future_use	1
reserved	2
section_length	12
transport_stream_id	16
reserved	2
version_number	5
current_next_indicator	1
section_number	8
last_section_number	8
original_network_id	16
reserved_future_use	8
for (i = 0; i < N; i++) {	
service_id	16
reserved_future_use	6
EIT_schedule_flag	1
EIT_present_following_flag	1
running_status	3
free_CA_mode	1
descriptors_loop_length	12
for (i = 0; i < M; i++) {	
descriptor()	
}	
}	
CRC_32	32

Tabulka 4.10: Syntax SDT tabulky

Kapitola 5

Hardwarové vybavení pro nahrávání dat

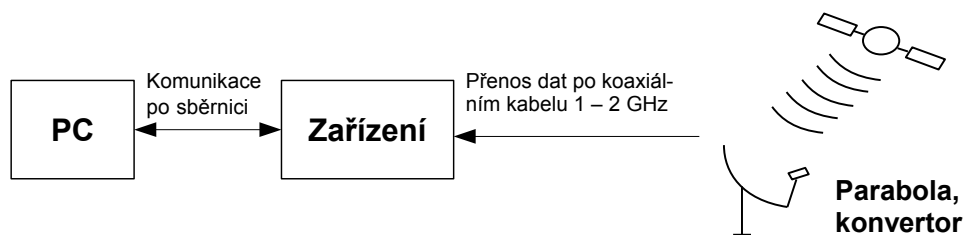
Základním stavebním blokem celého systému je nějaké zařízení, které nám umožňuje příjem MPEG2 transport stream datového toku v operačním systému Linux. Jelikož je žádoucí masivní nahrávání mnohojazyčných dat, rozhodl jsem se pořizovat data ze satelitů. Prvky které budeme potřebovat pro příjem jsou následující:

1. Zařízení pro získávání MPEG 2 datového toku,
2. Parabolická satelitní anténa,
3. Konvertor provádějící konverzi signálu o kmitočtu 10,7 až 12,5 GHz na signál cca 1 až 2 GHz, který je možné přenášet po koaxiálním kabelu do zařízení,
4. koaxiální kabel.

Podrobnější informace o problematice fyzického příjmu a přenosového systému pro příjem televize ze satelitu je možné nalézt např. v [9].

5.1 Výběr vhodného zařízení

Model řešeného problému je znázorněn na 5.1.



Obrázek 5.1: Model hardwarového vybavení

Stál jsem před problémem vybrat zdroj dat MPEG2 transportního datového toku. V době výběru byly možné dva způsoby jak toto provést:

- pomocí karty vsunuté do sběrnice PCI nebo
- pomocí zařízení na směrnicí USB.

Sběrnice USB byla velice záhy zavržena, protože potřebujeme masivní nahrávání dat, hardwarové zařízení, které by pracovalo přes sběrnici USB a umožňovalo přenášet do PC velký datový tok též není finančně dostupné. Dalším problémem by byla nutnost provozovat nahrávání na PC, které má integrovaný USB 2.0 kontrolér (na základní desce), to by způsobilo též prodražení. Rozhodl jsem se tedy vybrat nějaké zařízení do sběrnice PCI, dostupné karty se daly v té době rozdělit do dvou skupin:

1. Karty s vestavěným dekodérem MPEG 2 video a audia ve formátu MP2, po rozšíření o speciální CA modul, tyto karty umožňovaly i vkládání přístupových smart karet pro podmíněný přístup, disponovaly výstupem pro připojení televize (kompozitní TVout a S-Video TVout) a audio výstupem. Z hlediska možnosti masivního nahrávání nebyly příliš na výši. Důvodem bylo, že při přehrávání programu ze satelitu byla velká šíře přenosového kanálu věnována odesílání dekodovaných video dat přes sběrnici PCI do grafického adaptéru, když chtěl uživatel sledovat vysílání na monitoru PC. Toto chování bylo sice možné eliminovat, tak či tak dle specifikace produktů z této kategorie vyplývala malá přenosová rychlost pro přenos MPEG 2 transportního datového toku z PCI karty do PC. Cena karet tohoto typu se pohybovala v cenové relaci 5000 až 10000 Kč podle výbavy.
2. Redukovaná verze karet z předchozího odstavce. Karty byly “ořezány” tak, že obsahují pouze základní důležité komponenty, tedy Tuner, obvody pro korekci chyb na fyzické vrstvě standardu DVB, PCI rozhraní a napájecí obvod konvertoru na parabole. Funkce karet tohoto typu spočívá pouze v naladění určitého transpondéru (tj. vysílací satelitní kanál), a přenosu MPEG 2 transportního datového toku do aplikace v PC. Veškeré další zpracování, jako dekodování transport streamu, dekodování audio a video dat a prezentace se provádí softwarově v PC. Tento přístup je výhodný, HW část řešení je mnohonásobně zjednodušena, po sběrnici se přenáší data kódovaná, tedy pouze datový tok převzatý přímo ze satelitu. Karty s tímto přístupem ke zpracování satelitního vysílání se vyznačují možností vysokého datového toku mezi HW a PC, což umožňuje přenášet velké masivní množství řečových signálů. Další výborná vlastnost těchto karet je i jejich cena, která se pohybuje v rozmezí 1800 až 2800 Kč.

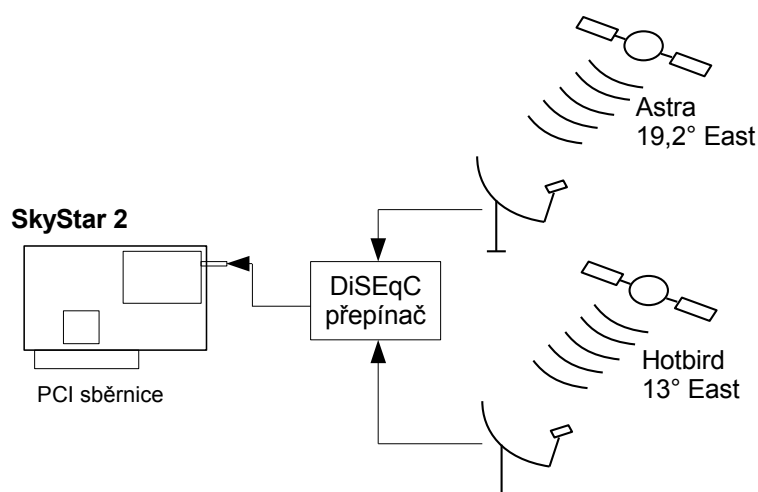
Z výše uvedených vlastností jednotlivých druhů karet jsem vybíral nejvhodnější. Zařízení z druhé skupiny je mnohem vhodnější, protože na jednu stranu pořizovat kartu s HW dekodérem MPEG2 obrazu by bylo trochu neúčelné, na druhou stranu nás zajímá průchodnost dat mezi HW a PC, kde karty druhého typu ze specifikace dominují.

Pro výběr z více karet typu 2 jsem stanovil jasná pravidla, prioritní je bezproblémová činnost v operačním systému Linux. Z karet, které byly v době výběru možné koupit, jednoznačně vycházela nejlépe karta **SkyStar2** od společnosti Technisat. Hlavní výhodné parametry:

- Ovladače pod operační systém Linux existují, jsou vyráběny společností B2C2 Inc.
- Ovladače pod operační systém Microsoft windows fungují v rámci možností, jsou vyráběny společností B2C2 Inc.
- Cenově dostupná, cca 2000 Kč.

- Krátce po koupi této karty se firma B2C2 přestává vydávat aktualizace ovladačů a dochází k převodu vývoje všech ovladačů na opensource skupinu LINUXTV.ORG, což dává i přístup ke zdrojovým kódům ovladače. Dochází k vylepšování vlastností a zkvalitňování ovladače. Ovladače pro jádro 2.6 jsou k dispozici.
- Karta disponuje HW filtrem MPEG 2 Transport stream datového toku, který spočívá v přenosu informací, z PC do HW, o PIDech, které se mají přenášet do PC. Karta bude od této chvíle přenášet do PC pouze žádané PIDy. Tento filtr je nutný protože standardní šíře jednoho transpondéru se pohybuje v rozmezí 22 až 32 Mb za sekundu, takový datový tok je velice problematické po sběrnici PCI přenášet.
- Umožňuje ovládání zařízení umístěných na koaxiálním kabelu mezi kartou a konvertorem (na parabole), pomocí rozhraní DiSEqC. V tomto systému budeme používat DiSEqC přepínač parabol.

Celková HW konfigurace bude tedy počítat s kartou SkyStar2, dvěma konvertory a parabolami, DiSEqC přepínačem parabol a propojovacího koaxiálního kabelu, jak je znázorněno na obrázku 5.2. V dalším textu stručně popíšu základní obsluhu karty SkyStar2 pomocí linuxtv.org ovladačů a základní parametry rozhraní DiSEqC.



Obrázek 5.2: Výsledná konfigurace hardwarové části nahráváče řečových dat

5.2 DiSEqC sběrnice

Tato sekce popisuje funkční specifikaci sběrnice pro ovládání zařízení (Digital Satellite Equipment Control - DiSEqC). DiSEqC je otevřený standard s dodatky, které jsou schváleny na základě průmyslových dohod.

Systém DiSEqC je komunikační sběrnice mezi satelitním přijímačem a periferními zařízeními, která používá pouze existující koaxiální kabel. DiSEqC může být integrován do zákaznických satelitních zařízení a nahrazuje všechny dříve používané analogové techniky (např. změna napětí, změna frekvence tónu) přepínání a další techniky, které potřebovaly navíc přídatné dráty vedle koaxiálního kabelu. Univerzální podřízený integrovaný obvod

podporuje řadu aplikací, od řízení linky až po identifikaci nadřazeného členu, který ho řídí. Hlavní výhody systému jsou:

- digitální systém, který je standardizován bez proprietárních příkazů,
- podporuje přepínání při příjmu více satelitů,
- zpětně kompatibilní s přepínáním 13/18V a tónem 22kHz,
- potenciál pro redukci zbytečně vyzářeného výkonu a tedy snížení ceny a zvýšení spolehlivosti,
- odstranění problému při přepínání zapříčiněnou nekompatibilitou jednotlivých komponent,
- jednodušší instalace přijímače, použití rozpoznání zařízení pomocí volitelné dvoucestné komunikace.

Koncept spočívá v rozšíření současné 22 kHz tónové signální metody, tedy minimalizováním nutnosti změn v tuneru či dekodéru přijímače a snadnější zpětné kompatibilitě. Od té doby co DiSEqC protokol obsahuje zpětnou signální cestu je nutné mnohem přesněji definovat impedanci na sběrnici, než dříve kdy se používala tónová metoda.

DiSEqC sběrnice se skládá z jednoho nadřazeného prvku (master) a jednoho či více podřízených prvků (slave), komunikace může být zahájena pouze nadřazeným prvkem. Tento princip zabraňuje nutnosti mít na straně nadřazeného prvku nějakou utilitu k neustálému monitorování sběrnice. Principiálně nadřazený prvek způsobuje přerušování existujícího 22 kHz tónu ke komunikaci s prvkem podřízeným.

Funkce podřízeného prvku bývá obvykle implementována do jednoduchého mikrokontroléru. Takovýto mikrokontrolér musí být schopen dekódovat informace vysílané po sběrnici, vybrat ty informace, které jsou pro něj určeny a provádět operace pro které je určen.

5.3 Ovládání karty SkyStar2

Zařízení pro ovládání DVB karet jsou podle [1] logicky organizovány v adresáři `/dev/dvb`. Každá karta připojená do systému má v tomto adresáři podadresář, který obsahuje speciální ovládací zařízení. Podadresáře mají názvy `Adapter0` až `AdapterN`, jejichž obsahem jsou následující ovládací zařízení:

- `audioN`
- `videoN`
- `frontendN`
- `netN`
- `demuxN`
- `caN`
- `dvrN`

kde N nabývá hodnot od 0 po počtu těchto jednotlivých zařízení na kartě. Zařízení `audioN` a `videoN` se používají v případě, že karta obsahuje hardwarový dekodér audio či video elementárních datových toků. V těchto případech jsou dostupná pomocí `audioN` a `videoN` přímo dekomprimovaná data pro přenos na zvukovou resp. grafickou kartu. `caN` se používá pokud karta obsahuje nějaké zařízení pro podmíněný přístup.

Pro naše použití jsou nejdůležitější zařízení `frontendN`, `demuxN` a `dvrN`. K ovládní tuneru a DiSEqC sběrnice se používá zařízení `frontendN` a k definici PIDů (tedy k ovládní HW filtru transport streamu) se používá `demuxN`. Zařízení `dvrN` se používá jako personální nahrávač elementárních datových toků. Postup práce s kartou `skystar2` je možné zjednodušeně popsat takto:

1. Otevření zařízení `frontend0`, nastavení parametru DiSEqC, nastavení transpondéru. Transpondér se nastavuje pomocí 3 hodnot, symbolová rychlost, frekvence transpondéru a polarizace transpondéru.
2. Otevření zařízení `demux0`, nastavení PIDů, které se mají nahrávat a zahájení činnosti filtru.
3. Otevření zařízení `dvr0`, čtení dat a jejich zápis do souboru na disku.
4. Pro ukončení je nutné zastavit činnost hardwarového filtru a provést uzavření všech otevřených deskriptorů.

Následující příklad demonstruje jednoduchý program pro nastavení a získání dat z karty `SkyStar2`. V programu je vypuštěna část kódu, kde se nastavuje konvertor na parabole, a zjišťuje frekvence tuneru na kartě. Důvod je ten, že tato část kódu je prostorově náročná a nepřehledná, cílem tohoto programu je předvést základní princip komunikace s kartou místo třech nepřehledných stran.

```
//demonstrační kód pro načtení 5-ti paketů
//PAT tabulky ze satelitu
....

static int fefd, dmxfd, dvrdev;
int result, ifreq;
static struct dvb_frontend_info fe_info;
struct dvb_frontend_parameters tuneto;
struct dmx_pes_filter_params pesfilter;
char buffer[5*188];

//otevření zařízení
fefd = open('/dev/dvb/adapter0/frontend0', O_RDWR | O_NONBLOCK);
result = ioctl(fefd, FE_GET_INFO, &fe_info); //zjištění informací
if (fe_info.type != FE_QPSK) //zjištění zdali se jedná o DVB-S zařízení
{ // chyba }
// nyní následuje kód pro nastavení parametrů konvertoru na parabole
// pomocí DiSEqC rozhraní, tento kód je příliš dlouhý a zbytečně by
// zneřehledňoval tuto demonstraci, výsledkem tohoto kódu je též
// frekvence tuneru, která je v proměnné ifreq.
```

```

tuneto.frequency = ifreq;
tuneto.inversion = INVERSION_AUTO;
tuneto.u.qpsk.symbol_rate = sr; // zde je nastaven symbol rate, parametr
                                // potřebný pro naladění transpondéru
tuneto.u.qpsk.fec_inner = FEC_AUTO;
ioctl(fefd, FE_SET_FRONTEND, &tuneto);
usleep(1000000) //prodleva před další komunikací s kartou

// otevření demux rozhraní
dmxfd = open('/dev/dvb/adapter0/demux0', O_RDWR);
ioctl(dmxfd, DMX_SET_BUFFER_SIZE, 10*188); //velikost bufferu
pesfilter.pid = 0; //nastavení PAT PIDu k nahrávání
pesfilter.input = DMX_IN_FRONTEND;
pesfilter.output = DMX_OUT_TS_TAP; // výstup v podobě TS paketů
pesfilter.pes_type = DMX_PES_OTHER;
pesfilter.flags = DMX_IMMEDIATE_START;
ioctl(dmxfd, DMX_SET_PES_FILTER, &pesfilter); // start nahrávání

//otevření souboru pro čtení dat
dmxfd = open('/dev/dvb/adapter0/dvr0', O_RDONLY);
read(dvrfd, buf, 5*188);

ioctl(dmxfd, DMX_STOP); // zastavení nahrávání
close(dmxfd);close(dvrfd);close(fefd); // uzavření deskriptorů

....

```


Kapitola 6

Aplikace pro nahrávání satelitních dat

Na aplikaci pro nahrávání dat jsou kladeny následující požadavky:

- nahrávání více audio a teletextových streamů současně,
- možnost používat více karet SkyStar 2,
- pomocí DiSEqC přepínače umožnit přepínat mezi více parabolami (na každé kartě),
- spouštět nahrávání plánovačem,
- umožnit další zpracování nahraného záznamu.

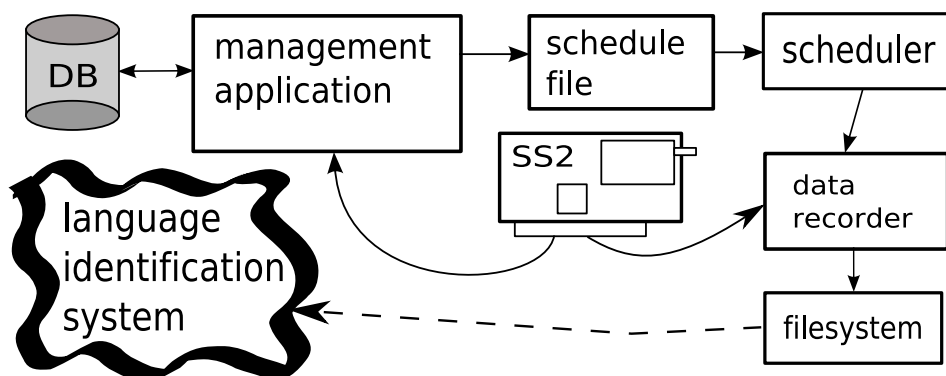
Blokové schéma systému je znázorněno na obrázku 6.1. Hlavní částí je data recorder, který komunikuje s kartou SkyStar2, provede naladění určitého transpondéru, definuje PIDy, které se mají nahrávat a ukládá data do souboru na disk. Informace o tom co kdy a odkud se má nahrávat je umístěno v plánovacím souboru. Plánovač je proces, který běží neustále, a když je to nutné (podle plánu nahrávání), zajišťuje spouštění a ukončování nahrávače dat. Po ukončení nahrávání provede spuštění skriptu, který umožňuje další zpracování nahraných dat.

Vstupní předpis pro nahrávač dat (plánovací soubor) je vytvářen buď aplikací pro správu, nebo je možné ho vytvořit ručně. Obsahuje informace potřebné pro nahrávání, jako je definice zařízení DVB karty, specifikace parametrů transpondéru, dále seznam PIDů a časové údaje kdy se má nahrávat.

6.1 Aplikace pro správu dat

Aplikace pro správu je databázová aplikace a má několik důležitých úkolů:

- **Skenování** – Umožňuje získávat informace o službách vysílaných ze satelitu a ukládá je do databáze.
- **Definice PIDů, které se mají nahrávat** – Jeden TS datový tok obsahuje více služeb, při nahrávání je tudíž možné nahrávat více kanálů naráz. Omezujícím faktorem jsou pouze vlastnosti HW filtru na SkyStar2 a datová propustnost sběrnice PCI. Označíme tedy v databázi elementární TS toky, které se budou nahrávat a data rekordér je bude nahrávat všechny naráz.



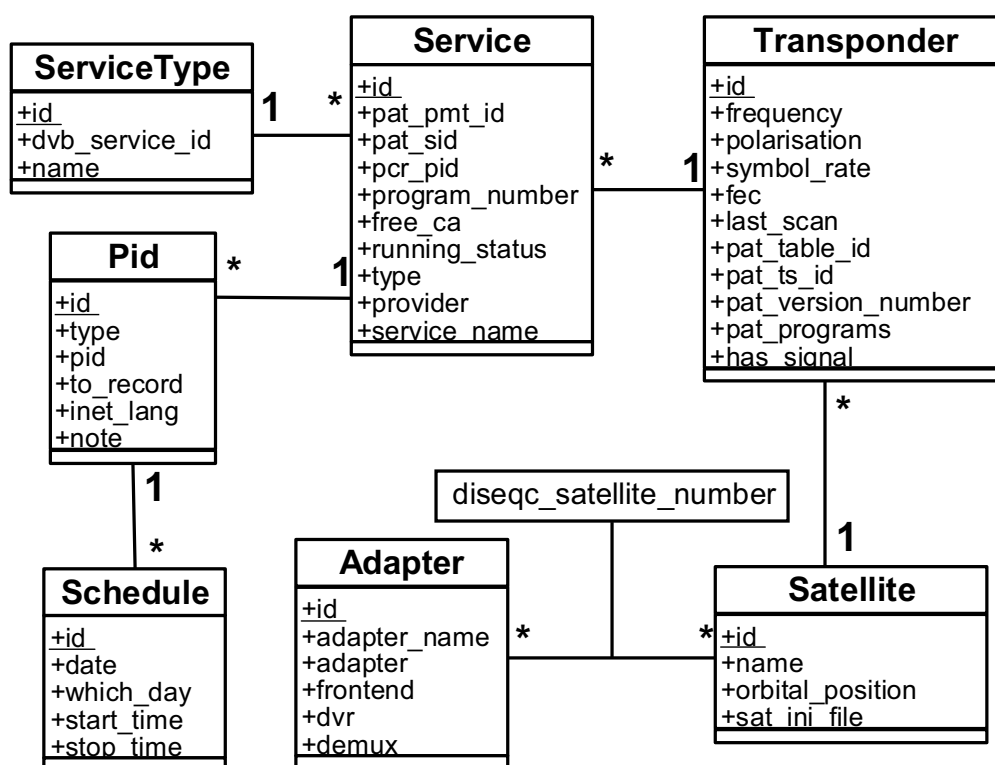
Obrázek 6.1: Schéma aplikace pro nahrávání dat

- **Tvorbu plánovacích souborů** – Grafická nástavba, která umožňuje komfortnějším způsobem vytvářet plánovací soubory.
- **Získávání dodatečných informací z internetu** – Systém bude umožňovat získávat dodatekové informace o jazycích služeb ze satelitu, pokud jsou vysílány. Dále bude umožňovat přidávat vlastní poznámky k jednotlivým PIDům služeb. Na internetu je možné získávat volně dostupné seznamy programů, které ze satelitů vysílají i s detailními a mnohdy explicitně odvozenými informacemi o jazycích jakými jsou programy vysílány.

Nutná součást aplikace je vhodné úložiště dat. Model tohoto úložiště je znázorněn na obrázku 6.2. Vysvětlení funkce jednotlivých entit je sumarizováno v následujících odrážkách.

- Entita **Adapter** obsahuje název zařízení a definici všech zařízení, které je potřebné k práci s adaptérem, tj. hlavně frontend, demux, dvr zařízení aj.
- **Satellite** definuje název satelitu, pozici na orbitu a název souboru s definicí transpondérů pro daný satelit. Jelikož je mezi entitou adapter a satellite vztah Many-to-Many s atributem, při konverzi na schéma databáze je nutné vytvořit další tabulku, která umožňuje tento vztah fyzicky realizovat.
- **Transponder** definuje všechny parametry nutné pro naladění transpondéru, to jsou frekvence, symbolová rychlost, polarizace atd. Dále obsahuje informace zjištěné z PAT tabulky tohoto transpondéru.
- **Service** obsahuje informace společné pro službu, které jsou zjištěné z tabulek PAT, PMT a SDT, je to název služby, zprostředkovatel služby, PID PMT tabulky, SID služby atd., dále používá číselník **cis_service_type**, který obsahuje všechny dostupné typy služeb.

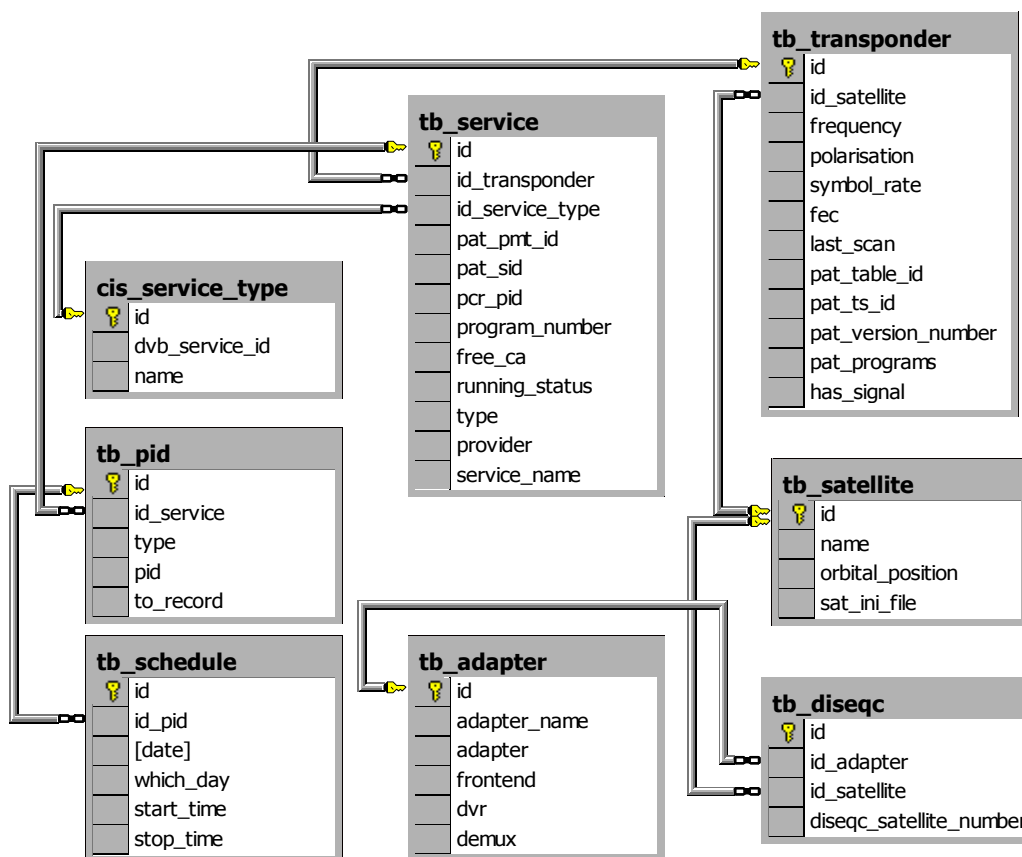
- **Pid** obsahuje informace o PIDEch přidružených k službě, jdou to vlastní PID hodnoty, typ elementárního datového toku, pole pro uložení jazyku z internetu a vlastní poznámku.
- Entita **Schedule** obsahuje definici času, odkdy pokud je nutné provádět nahrávání jednotlivých PIDů. Dále obsahuje definici, který den či které dny v týdnu se budou data nahrávat.



Obrázek 6.2: Model úložiště dat

Model nám umožňuje definovat přesně, který adaptér je přes přepínače připojen ke kterým parabolám (posléze k satelitům), a informace o všech transpondérech, službách a PIDEch, které je možné na všech satelitech naladit. V databázi jsou též položky pro doplňující informace, které se budou buď zadávat ručně nebo se budou automatizovaně zjišťovat z internetu. Každý PID je možné zařadit k nahrávání.

Tento model je nutné konvertovat na schéma databáze. Jako databáze byl vybrán MySQL server, který bude běžet na stejném PC jako data rekordér a celá aplikace. Schéma této databáze je znázorněno na obrázku 6.3. Pro zjednodušení práce s databází, byl též zvolen typ databáze a typ tabulek umožňující funkci “ON DELETE CASCADE”, jinými slovy při operaci smazání např. transpondéru, dochází též k vymazání všech jeho služeb a PIDů k službám přidruženým a to vše automaticky na úrovni systému pro řízení báze dat.



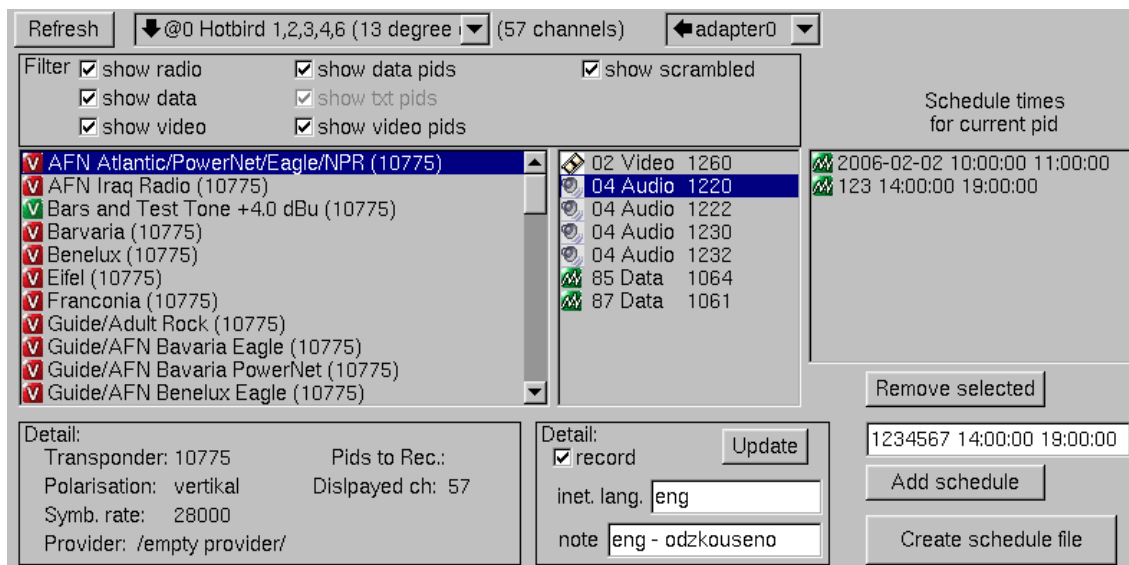
Obrázek 6.3: Schéma databáze úložiště dat

6.1.1 Modul pro práci s databází

Hlavní částí aplikace pro správu je rozhraní pro zobrazování obsahu databáze, které je zobrazeno na obrázku 6.5. Dialog je navržen tak, aby přehledně a logicky zobrazoval obsah databáze. Celý dialog je možné rozdělit na několik základních částí.

- Horní panel,
- Panel pro filtrování zobrazovaného obsahu,
- Seznam zobrazující všechny služby včetně doplňujících informací (dole),
- Seznam všech PIDů, které jsou přidruženy k vybrané službě včetně detailních informací (dole),
- Seznam definic nahrávání pro vybraný PID, včetně ovládání tohoto seznamu.

Horní panel slouží k vybrání karty v systému se kterou se bude pracovat a posléze k vybrání aktuálního satelitu. Tato část je pevně spjata s tabulkami *tb_adapter*, *tb_diseqc* a *tb_satellite*. Všechny důležité informace jsou zobrazeny, jako název adaptéru, název družice a DiSEqC přepínací hodnota, která je uvedena před názvem družice. Tento Panel slouží



Obrázek 6.4: Rozhraní aplikace pro zobrazování obsahu databáze

jako základní filtr pro zobrazování kanálů v následujících panelech. Tlačítko Refresh slouží k znovunačtení seznamu kanálů.

Panel pro filtrování zobrazovaného obsahu provádí další filtraci programů a PIDů. Pro filtraci programů slouží položky *show radio*, *show data*, *show video* a *show scrambled*. Pro filtraci PIDů všech programů slouží zbylé volby *show data pids*, *show teletext pids* a *show video pids*.

Seznam služeb obsahuje všechny služby, které jsou filtrovány nastavením v předchozích panelech. Každá položka seznamu je vybavena ikonou, která definuje druh služby a barvou ikony, která specifikuje to zda je služba šifrovaná. V panelu pod seznamem jsou doplňující informace k vybrané službě a to definice transpondéru na kterém je možné službu nahrávat a provozovatele služby (tj. informace z tabulky SDT), pokud je vysílán. Tato část dialogu koresponduje s informacemi z tabulek *tb.transponder* a *tb.service*.

Seznam PIDů obsahuje všechny PIDy přidružené k právě zvolené službě po aplikaci filtru zvoleného v panelu filtr. U každého PIDu je uvedena jeho hodnota a typ. Při vybrání nějakého pidu jsou v panelu pod tímto seznamem zobrazeny tyto informace:

- Zaškrávací políčko *record* – slouží k definici, zda se tento PID bude nahrávat.
- Položka *inet lang* – slouží k zobrazení popř. změně jazyku elementárního datového toku, který byl importován z internetu.
- Položka *note* – slouží k definici libovolné poznámky k elementárnímu datovému toku.

Jelikož dochází v této sekci k manipulaci s obsahem databáze je tu přítomno tlačítko, kterým se provádí uložení všech změn provedených v této sekci zpět do databáze. Při stisknutí tlačítka dojde k uložení všech třech informací z tohoto panelu naráz. Tato část pracuje s daty z tabulky *tb.pid*.

Posledním logickým celkem je *Seznam definic nahrávání pro vybraný PID*. Při označení PIDu služby jsou zobrazeny všechny naplánované časy spojené s tímto PIDem. Je možné tyto záznamy odebírat (označením a zmáčknutím tlačítka *Remove selected*) a přidávat vepsáním záznamu do políčka a stisknutím tlačítka *Add schedule*.

Existují dva způsoby jak zadat plán pro nahrávání. Buď specifikovat den (popř. dny) v týdnu, kdy nahrávat nebo zadat přesné datum, kdy nahrávat. Syntax jednotlivých zápisů je zobrazena v tabulce 6.1. Při manipulaci se seznamem dochází přímo k manipulaci s databází, dochází k mazání a vkládání dat do tabulky *tb_schedule*.

Syntax	Význam
<den> <čas od> <čas do>	Definice pro nahrávání přesně v jeden den, který je specifikován ve formátu YYYY-MM-DD.
2006-02-02 11:00:00 12:00:00	Příklad
<dny> <čas od> <čas do>	Definice pro nahrávání v určité dny v týdnu, dny jsou definovány pomocí čísel 1 – 7, přičemž 1 je pondělí a 7 odpovídá neděli.
12367 11:00:00 12:00:00	Příklad

Tabulka 6.1: Syntax definice časů nahrávání

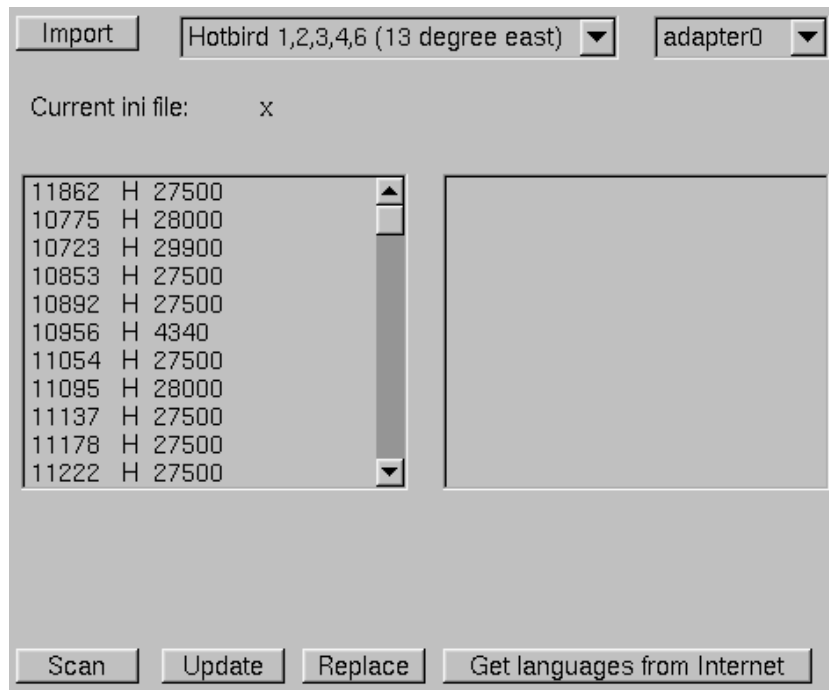
Poslední částí aplikace je tlačítko pro vytvoření plánovacího souboru. Po stisknutí je v adresáři spustitelného souboru aplikace vytvořen soubor, který přesně reflektuje formát, který je popsán v odstavci 6.2. Přitom je brán zřetel na vybranou kartu, DiSEqC přepínač aktivní družice a aktivní transpondér. Tyto hodnoty jsou do souboru uloženy, vedle nich všechny PIDy, které se mají z tohoto transpondéru nahrávat spolu s definicemi kdy nahrávat.

6.1.2 Modul pro skenování satelitu

Hlavní části tohoto modulu jsou analyzátoři tabulek PAT, PMT a SDT. Dále je nutná utilita pro získávání tabulek ze satelitu. Činnost modulu je možné popsat v těchto krocích:

- Výběr karty a satelitu, které budou použity pro skenování. Výběr se provádí obdobně jako u předchozího modulu.
- Aplikace obsahuje tlačítko *Import*, které umožňuje přidat z externího souboru do databáze transpondéry, které se budou skenovat. Jméno externího souboru je zjištěno v tabulce *tb_satellite* a je zobrazeno na formuláři.
- Všechny transpondéry, které se budou skenovat, se zobrazí v levém seznamu.
- Stisknutím tlačítka *Scan* se zahájí proces skenování.
- Po dokončení se v druhém panelu objeví názvy všech služeb, které se podařilo oskenovat, pomocí tlačítek *Update* resp. *Replace* je možné provést aktualizaci resp. přepsání naskenovaných dat do databáze.

Proces skenování probíhá následovně, pro všechny transpondéry je nejprve ze satelitu nahrána tabulka PAT a SDT a je analyzován jejich obsah. Poté je podle informací z tabulky PAT provedeno nahrávání všech PMT PIDů a provedena analýza jejich obsahů. Nyní je



Obrázek 6.5: Rozhraní aplikace pro skenování satelitu

vytvořena temporální datová struktura, která obsahuje všechny potřebné informace o všech službách na aktuálním transpondéru a o všech jejich PIDech. Při opakování celého procesu pro všechny transpondéry satelitu se vytváří kolekce těchto struktur. Po dokončení jsou z této kolekce vypsány názvy všech služeb do druhého seznamu na formuláři.

Pomocí tlačítka *Update* je tato datová struktura předána funkci pro aktualizaci obsahu databáze, která funguje následovně. Pro jednotlivé transpondéry a již existující služby provádí změnu záznamů v databázi na nově zjištěné (např. přibyl jeden audio elementární datový tok) což má za následek, že se neztrácí informace o tom, kdy se má jaký PID nahrávat a popř. informace o jazyku a poznámka. Pokud se stiskne tlačítko *Replace* postup je obdobný, ale před tím než se začnou do databáze ukládat data o jednotlivých službách se provede vymazání všech služeb aktuálního transpondéru. Dojde tedy k “vyčištění” databáze před uložením nových dat, což má za následek ztrátu informací o nahrávání a poznámkách elementárních dat. toků.

Poslední neméně důležitou částí tohoto modulu je import jazyků z internetu. Vstup se očekává ve speciálním formátu, který je uveden na obrázku 6.6. Takovýto výstup produkuje skript *lyngsat_parse.py* přímo z HTML stránky, která je mnoha uživateli udržována na serveru <http://www.lyngsat.com>. Soubor je zpracováván jednoduchým konečným automatem, který vždy zjistí zdali je daný transpondér přítomný v databázi, pokud ano zjistí kolekci párů PID - jazyk. Provede aktualizaci databáze podle této kolekce a přejde na další transpondér. Veškeré data ukládá do tabulky *tb_pid* do datové položky *inet.lang*.

Na výše uvedené stránce jsou seznamy všech transpondérů, na kterých se vysílá. Vždy jsou uvedeni všichni provozovatelé daného transpondéru, přičemž některé programy jsou šifrované některé nikoliv. Velkou výhodou je že seznam uvádí parametry všech volně šířených programů ať jsou vysílány na transpondéru částečně šifrovaném či nikoliv.

```
DVB:<frekvence>*<symbolová rychlost>*<polarizace>
>><název služby 1>
>>>><pid 1> <jazyk>
>>>><pid 2> <jazyk>
>><název služby 2>
>>>><pid 1> <jazyk>
>>>><pid 2> <jazyk>
>>>><pid 3> <jazyk>
DVB:<frekvence>*<symbolová rychlost>*<polarizace>
>><název služby 3>
>>>><pid 1> <jazyk>
>>>><pid 2> <jazyk>
.....
```

Obrázek 6.6: Syntax souboru pro import jazyků

6.2 Plánovací soubor (schedule file)

Plánovací soubor nese všechny důležité informace, které jsou nutné pro nahrávání multi-mediálních dat ze satelitu. Obvykle je generován aplikací pro správu, avšak má pevně definovanou syntax a sémantiku a velice jednoduchou strukturu, takže je možné ho vytvářet i ručně. Plánovací soubor je využitý ve dvou komponentách celého systému, je to nahrávač dat a plánovač.

Pro nahrávač jsou v souboru definovány všechny informace nutné k identifikaci karty, která se použije, definice transpondéru, který je nutno naladit, a seznam PIDů, které se budou nahrávat. Pro plánovač soubor nese zásadní časové informace o tom, kdy se bude jaký PID nahrávat.

Syntax plánovacího souboru je znázorněna na obrázku 6.7. Pro naladění transpondéru je použita definice `\tune`, frekvenci transpondéru je nutno nastavit v MHz, symbolovou rychlost v kS/s (kilosymbol za sekundu) a polarizace se nastavuje na písmena 'V' pro vertikální a 'H' pro horizontální. Přepínací hodnota DiSEqC specifikuje číslo paraboly, která se použije pro naladění transpondéru. Toto číslo je nahrávačem dat posláno pomocí DiSEqC příkazů do přepínače, který je umístěn na koaxiálním kabelu mezi kartou Skystar2 a konvertory na parabole.


```

#
# komentář => celý řádek se ignoruje!
\tune <frekvence transpondéru>*<symbolová rychlost>*<polarizace>
\diseqc <přepínací hodnota DiSEqC>
#
\fedevice <identifikace zařízení>
\dmxdevice <identifikace zařízení>
\dvrdevice <identifikace zařízení>
#
\postprocess <název skriptu následného zpracování>
#
\pid <pid> ‘‘<identif>’’ <definice dnu(ů) kdy nahrávat> <čas od> <čas do>
#
#

```

Obrázek 6.7: Syntax plánovacího souboru

Další důležitou položkou v plánovacím souboru jsou ovládací zařízení karty.

- `\fedevice` – zařízení ovládající DiSEqC sběrnici a tuner karty, nejčastěji `/dev/dvb/adapter0/frontend0`,
- `\dmxdevice` – zařízení ovládající HW filtr karty, nejčastěji `/dev/dvb/adapter0/demux0`,
- `\dvrdevice` – zařízení pro nahrávání dat, nejčastěji `/dev/dvb/adapter0/dvr0`.

Skript následného zpracování je spouštěn plánovačem po ukončení nahrávání. V každém plánovacím souboru je možné definovat jiný skript, ten je volán s pevně danými parametry, ty jsou specifikovány v sekci 6.3.

Nyní se dostáváme ke specifikaci PIDu, který je nutné nahrát. Datová položka `pid` specifikuje PID elementárního datového toku v dekadické podobě. Položka `identif` určuje identifikátor PIDu ve formátu:

```
<typ pidu>-<nějaký řetězec(název stanice)>-<jazyk získaný z internetu>
```

Definice dnů, kdy se má nahrávat je možné provádět dvěma způsoby. Buď je možné provést nastavení na jeden přesný den ve formátu `YYYY-MM-DD`, nebo je možné specifikovat dny v týdnu (ve formátu `EVERY <jednotlivá čísla dne>`), kdy se bude nahrávání spouštět. Čas spuštění a čas zastavení se uvádí ve formátu `HH:MM:SS`.

Nyní následuje jednoduchý příklad zápisu plánovacího souboru. Jedná se o transpondér 57 na satelitu Astra 19,2°, pro definici časů, je demonstrováno nastavení pro jeden jediný den i pro opakování v přesně specifikované dny v týdnu.

```

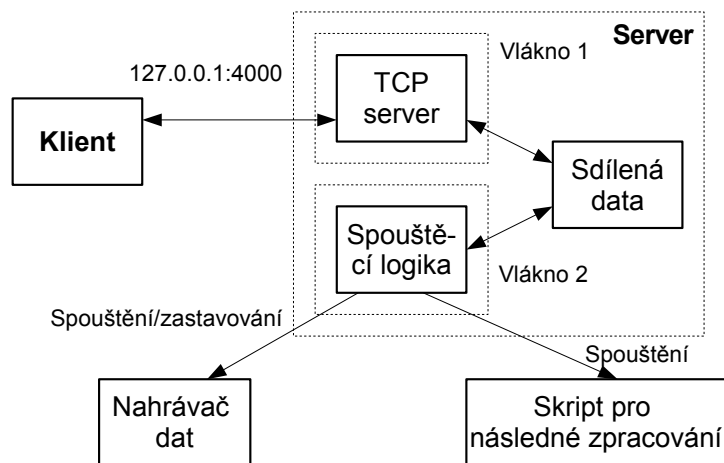
#
\tune 10832*22000*H
\diseqc 1
#
\fedevice /dev/dvb/adapter0/frontend0
\dmxdevice /dev/dvb/adapter0/demux0
\dvrdevice /dev/dvb/adapter0/dvr0
#
\postprocess postprocess
#
\pid 73 'a-Radio_Marya-Po' 2007-04-12 11:00:00 12:00:00
\pid 64 'a-Radio_Horeb-G' EVERY 1234567 09:00:00 09:30:00
\pid 73 'a-Radio_Marya-Po' EVERY 12345 14:00:00 17:00:00
\pid 40 'a-24_CZ-Cz' EVERY 67 08:00:00 12:00:00
#

```

Obrázek 6.8: Příklad plánovacího souboru

6.3 Plánovač

Hlavním úkolem plánovače je spuštění a ukončování nahrávače dat v přesně definované časy, které jsou specifikovány v plánovacím souboru. Plánovač je založen na architektuře klient-server a je paralelní aplikací. Schéma je znázorněno na obrázku 6.9. Klient slouží

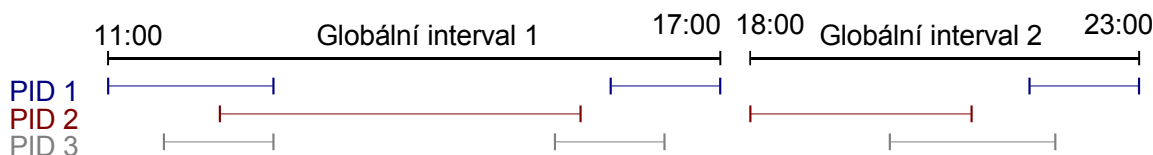


Obrázek 6.9: Schéma plánovače

ke kompletnímu řízení plánovače pomocí příkazů, které jsou posílány přes TCP lokální smyčku. Vlastní plánovač je tvořen dvěma vlákny a dvěma sdílenými datovými strukturami. Tyto datové struktury jsou stráženy pomocí jednoho mutexu, aby bylo zajištěno to že se budou vždy nacházet v konzistentním stavu.

Funkce celého plánovače je možné rozdělit do dvou fází. První fáze definuje co bude plánovač nahrávat, jedná se tedy o poslání příkazu o přidání plánovacího souboru přes lokální smyčku. Tato akce způsobí otevření plánovacího souboru, načtení veškerých důležitých dat a uložení do první datové struktury. Tuto fázi provádí klient a TCP server. Druhá fáze spočívá ve vygenerování druhé datové struktury, která přesně řídí spouštění a zastavování nahrávače dat. Tuto fázi zajišťuje Spouštěcí logika.

Plánovač je navržen tak, aby bylo možné provádět nahrávání dat z více PIDů v různou dobu naráz. Tento požadavek má za následek existenci dvou sdílených datových struktur. Jeden plánovací soubor může obsahovat k nahrávání více PIDů na jednom transpondéru, to má za následek nutnost vytvoření globálních intervalů pro nahrávání určitého transpondéru. Každý tento interval musí nést informace o PIDech, které se musí nahrávat i délky jednotlivých úseků a časovou diferenci od začátku vůči celému intervalu. Jednoduše řečeno, každý globální interval sebou nese definici všech úseků. Každý úsek je definován PIDem, délkou a vzdáleností od počátku intervalu. Demonstrace je provedena na obrázku 6.10, kde jsou přehledně zobrazeny jednotlivé úseky pro nahrávání.



Obrázek 6.10: Význam globálních intervalů

Každý řádek obsahuje úseky, kdy má dojít k nahrávání daného PIDu. Celé toto schéma je možné rozdělit do dvou globálních intervalů, jak je naznačeno. Každý globální interval musí obsahovat seznam všech úseků s všemi nutnými informacemi (délka, vzdálenost od počátku a PID).

Jako algoritmus pro rozdělení všech nahrávaných úseků do globálních intervalů byl zvolen následující klasifikační postup. Vstupem algoritmu je konečný počet úseků.

1. První úsek je automaticky vložen do první výstupní skupiny.
2. Pro další úsek zkontroluj všechny výstupní skupiny.
3. Pokud úsek překrývá nějaký úsek z výstupní skupiny, přidej ho do této skupiny, pokud ne vytvoř novou výstupní skupinu.
4. 2,3 opakuj pro všechny vstupní úseky.

Výstupem tohoto algoritmu je seznam skupin úseků. Pro konverzi na globální intervaly je nutné pro každou skupinu zjistit začátek a konec intervalu a identifikaci úseků transformovat na délku, PID a diferenci od počátku intervalu.

Výše popsaná funkčnost plánovače je využita pro konverze z první datové struktury do druhé datové struktury. Tato procedura se provádí velice často, vždy když skončí nějaké nahrávání je provedeno "čištění" datové struktury (odstranění všech záznamů, kde neprobíhá nahrávání) a opětovnému vytvoření datové struktury. Tímto přístupem je zajištěno,

že pokud bude požadováno přidání dalšího plánovacího souboru bude hned po ukončení aktivního nahrávání toto přidání reflektováno.

Pokud dojde k situaci, že bude požadováno nahrávání ze dvou plánovacích souborů naráz v určitém čase, bude tato situace řešena ukončením nahrávaného intervalu a začátkem nového intervalu. Provádět posuny začátku nahrávání není příliš vhodné, protože na začátek nahrávání jednotlivých globálních intervalů jsou vázány všechny úseky v něm obsažené. Došlo by ke značné deformaci záznamů.

Spouštěcí logika má za úkol mimo výše zmíněného neustálého přepřelánování též spouštění nahrávače dat, ukončování nahrávače dat a spouštění skriptu pro následné zpracování. Spouštění se provádí pomocí příkazů `fork` a `exec`, ukončování se provádí pomocí zaslání signálu `SIGINT`. Při startování nahrávače dat je reflektován formát spouštění, který je uveden v sekci 6.4. Pro startování skriptu pro následné zpracování je nutné předávat důležité parametry pro tvorbu výsledných nahrávek. První parametr specifikuje cestu k souboru s nahraným transportním datovým tokem. Další parametry specifikují, které PIDy (nahrávek) se mají z transportního toku extrahovat. Pro definici každé nahrávky jsou použity 3 parametry, PID, komentář k tomuto PIDu a vymezení intervalu nahrávky ve formě dvou čísel. První číslo specifikuje vzdálenost nahrávky od začátku datového toku v sekundách a druhé délku v sekundách. Pomocí tohoto principu skript pro následné zpracování produkuje záznamy, které přesně korespondují s naplánovanými úseky pro nahrávání.

Serverová část plánovač se spouští pomocí *sched*, klient pomocí *schcl*. Klient na jedno spuštění odešle severu jeden příkaz, který je nutno specifikovat jako první parametr příkazu. Pokud má příkaz parametr je nutné ho zadat jako druhý parametr na příkazové řádce. V tabulce 6.2 je seznam všech příkazů a parametrů, které je možné odeslat serveru spolu s krátkým popisem funkce.

Příkaz, parametry	Funkce
<code>list</code>	Vypíše obecný seznam právě používaných plánovacích souborů.
<code>detlist</code>	Vypíše detailní seznam právě používaných plánovacích souborů.
<code>term</code>	Provede ukončení plánovače.
<code>add <soubor></code>	Provede zařazení plánovacího souboru do první datové struktury.
<code>rem <číslo></code>	Provede vymazání plánovacího souboru z první datové struktury, číslo plánovacího souboru je specifikováno ve výpisu příkazu <code>detlist</code>
<code>actlist</code>	Vypíše právě naplánované nahrávání.
<code>kill <číslo></code>	Provede násilné ukončení probíhajícího nahrávání, číslo je specifikováno u výpisu <code>actlist</code> .

Tabulka 6.2: Příkazy plánovače

6.4 Nahrávač dat

Aplikace pro nahrávání dat principiálně odpovídá příkladu, který je uveden na konci předchozí kapitoly. Aplikaci je nutné předat jako první parametr cestu k plánovacímu souboru a jako druhý parametr definovat výstupní soubor, kam se budou nahrávat data.

Aplikace provede zjištění nutných dat z plánovacího souboru, naladí transpondér a definuje PIDy, které se budou nahrávat. Poté provádí ukládání dat do výstupního souboru. Činnost se ukončuje výhradně zasláním signálu SIGINT, což způsobí ukončení nahrávání, zavření všech podpůrných deskriptorů zařízení a ukončení nahrávače.

6.5 Skript pro následné zpracování

Tento skript je možné použít k mnoha akcím, které je nutné provést s nahraným transportním datovým tokem. V současnou chvíli je použit pro převod MPEG2 transport streamu na raw audio soubor s přesně nastavenými parametry. Tyto parametry jsou vstupní soubor s transport streamem a pro každý PID je specifikována trojice parametrů pro zpracování. Jsou to PID hodnota, identifikátor (z plánovacího souboru) a vymezení intervalu, který je pro daný PID relevantní. Detailní popis je uveden na konci sekce [6.3](#).

Skript je možné definovat v jednotlivých plánovacích souborech do budoucna se počítá se zpracováním teletextu či obrazových dat, je možné naprogramovat jakoukoliv funkčnost od pouhého převedení dat mezi formáty až např. po provedení segmentace zvukových či obrazových dat.

Kapitola 7

Závěr

Hlavním cílem této práce bylo najít zdroj pro získávání kvalitních mnohojazykových řečových nahrávek a umožnit automatizované nahrávání. Tyto data se používají pro trénování systémů pro identifikaci jazyka. Po zvolení zdroje řečových dat v podobě systému DVB-S bylo nutné kompletně prostudovat celý systém DVB včetně jeho datové vrstvy, MPEG2 transport stream.

Další fází bylo vybrání kvalitního prostředku pro získávání dat ze systému DVB a rozbor komunikace s tímto prostředkem. PCI karta SkyStar2 byla zvolena díky její bezproblémové funkci v operačním systému Linux a díky ovladačům v otevřené podobě. V této chvíli byly zjištěny všechny informace pro návrh architektury a implementaci aplikace pro nahrávání řečových dat.

Aplikace byla strukturně navržena z několika ucelených částí a tyto části implementovány. Dosavadní testování aplikace naznačuje, že vhodným výběrem stanic je možné získávat velice kvalitní řečové nahrávky, jedná se zejména o stanice, kde jsou vysílány zprávy či publicistika ve kterých je minimálně zastoupena řeč v jiných jazycích. Je tudíž velmi pravděpodobné že se systém bude dále hojně využívat pro zdokonalování systému pro identifikaci jazyka.

Můj osobní přínos z vypracování této práce spočívá v rozboru relevantních částí normy systému DVB a jeho datové vrstvy, pomocí něhož jsem byl schopen provést návrh celého systému. Při realizaci jsem používal různé techniky a různé programovací jazyky, jako jsou C++, Python a SQL pro komunikaci s databázovým serverem.

Možné směry dalšího vývoje jsou:

- Rozšíření nahrávání o videosekvence a teletextová data. Videosekvence mohou být dále použity pro systémy zpracování obrazu, teletextové informace mohou být využity jako podpora trénování v oblasti zpracování řeči.
- Přejít na nově se prosazující systém DVB-S2. Ten by měl být při zachování ovládacího rozhraní velice jednoduchý.
- Přeprogramování skenovací části aplikace na provoz z příkazové řádky. To by umožnilo automatizovat skenování satelitu a zjišťování dodatečných informací z internetu.
- Doimplementovat podporu nahrávání z internetových zdrojů. Aplikace by tedy zvládala záznam jak satelitního tak internetového vysílání.

Literatura

- [1] Dr. Marcus O.C. Metzler Dr. Ralph J.K. Metzler. Linux dvb api version 3. <http://linuxtv.org/downloads/linux-dvb-api-1.0.0.pdf>, 2003. Dokument dostupný – Květen 2007.
- [2] International Organization for Standardisation. *Generic coding of moving pictures and associated audio information: Audio*. ISO, 1994. ISO/IEC 13818-3.
- [3] DVB group. Framing structure, channel coding and modulation for digital terrestrial television, June 1996. DVB DOCUMENT A012.
- [4] DVB group. Digital Video Broadcasting (DVB); Framing structure, channel encoding and modulation for satellite services, June 1999. DVB DOCUMENT A040.
- [5] DVB group. Digital broadcasting systems for television sound and data services; framing structures, channel coding and modulation for cable systems, March 1998. DVB DOCUMENT A035.
- [6] European Telecommunications Standards Institute. Specification for service information (si) in dvb systems, December 2005.
- [7] International Telecommunication Union. *Audiovisual and multimedia systems - Transmission, multiplexing and synchronization*. ITU, 1996. ITU 222.0.
- [8] International Telecommunication Union. *Audiovisual and multimedia systems - Generic coding of moving pictures and associated audio information: Video*. ITU, 1999. ITU262.0.
- [9] Ing. Vladimír Vít. *Základny televizní techniky*. SNTL, 1987.