



# **ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE**

**Fakulta elektrotechnická**

Katedra telekomunikační techniky

## **“Webmike“ – zařízení pro vzdálený poslech prostřednictvím IP protokolu**

**Květen 2013**

**Student:**

**Bc. Miroslav Bašta**

**Vedoucí práce:**

**Ing. Pavel Troller, CSc.**

## **Čestné prohlášení**

Prohlašuji, že jsem zadanou diplomovou práci zpracoval sám s přispěním vedoucího práce a používal jsem pouze literaturu v práci uvedenou. Dále prohlašuji, že nemám námitek proti půjčování nebo zveřejňování mé diplomové práce nebo její části se souhlasem katedry.

Datum: 9.5.2013

.....

podpis diplomanta

Zde bude zadání diplomové práce.

## **Poděkování**

Rád bych poděkoval panu Ing. Pavlu Trollerovi, CSc. za čas, vedení a podporu, které mi poskytoval při tvorbě této diplomové práce.

**Anotace:**

Tato diplomová práce se zabývá tvorbou zařízení umožňující vzdálený poslech prostřednictvím IP protokolu. Zařízení je vybudováno na mikroplatformě Raspberry Pi a postaveno na linuxové distribuci Raspbian. Jádro práce spočívá v použití a konfiguraci vhodných linuxových aplikací, jedná se především o VoIPovou ústřednu Asterisk, streamovací server IceCast a webový server Apache.

**Klíčová slova:**

IP, Raspberry Pi, VoIP, Asterisk, stream, IceCast, Apache, HTML5

**Summary:**

The aim of this thesis is to build a device that enables long-distance transmission of live sound with the use of IP protocol. The device is built on Raspberry Pi platform using Raspbian Linux distribution. The core of this thesis lies in use and configuration of suitable Linux applications into the Raspbian distribution, namely VoIP Asterisk Exchange, IceCast stream server and Apache web server.

**Index terms:**

IP, Raspberry Pi, VoIP, Asterisk, stream, IceCast, Apache, HTML5

# Obsah

<b>1. Úvod</b>	<b>9</b>
<b>2. Přenos zvuku prostřednictvím IP sítě</b>	<b>10</b>
2.1 Struktura počítačové sítě	10
2.2 VoIP	11
2.2.1 Zpracování hlasového signálu	11
2.2.1.1 Vzorkování	12
2.2.1.2 Kvantování	12
2.2.1.3 Kódování	13
2.2.2 Přenos dat	13
2.2.2.1 RTP	13
2.2.2.2 UDP	14
2.2.3 Přenos signalizace	14
2.2.3.1 TCP	14
2.2.4 Implementace VoIP	15
2.2.4.1 H.323	15
2.2.4.2 SIP	17
2.2.5 SW klienti IP telefonie	19
2.3 Streamování zvuku	19
2.3.1 Jednocestný stream	19
2.3.2 Dvoucestný stream – WebRTC	20
2.3.3 Zpracování hlasového signálu	20
2.3.4 Přenos dat	21
2.3.5 Přenos signalizace	21
<b>3. Hardwarové řešení</b>	<b>22</b>
3.1 Základní hardware	22
3.1.1 Procesor a RAM	23
3.1.2 Vstupní a výstupní periferie	23
3.1.3 Operační systém	23
3.1.4 Další možnosti řešení	23

3.2	Hardware zpracovávající zvuk	25
3.2.1	USB zvuková karta	25
3.2.2	Mixážní pult/předzesilovač	26
3.2.3	Mikrofon	26
3.2.3.1	Uhlíkový	26
3.2.3.2	Piezoelektrický	26
3.2.2.3	Dynamický	26
3.2.3.4	Páskový	27
3.2.3.5	Kondenzátorový	27
3.2.3.6	Elektretový	27
<b>4.</b>	<b>Realizace</b>	<b>28</b>
4.1	Raspbian	28
4.1.1	Instalace	28
4.1.2	Konfigurace	28
4.2	Vytvoření streamu	29
4.2.1	DarkIce	30
4.2.1.1	Instalace	30
4.2.1.2	Konfigurace	31
4.2.2	IceCast	32
4.2.2.1	Instalace	32
4.2.2.2	Konfigurace	32
4.3	SIP/Asterisk	33
4.3.1	Instalace	33
4.3.2	Konfigurace	33
4.4	HTML5	35
4.4.1	HTTP server	35
4.4.2	Přehrávač	35
4.4.3	Webové stránky	36
<b>5.</b>	<b>Závěr</b>	<b>37</b>
<b>6.</b>	<b>Seznam literatury</b>	<b>38</b>
<b>7.</b>	<b>Elektronické Přílohy</b>	<b>40</b>



# 1. Úvod

Cílem této práce je vytvořit zařízení přenášející živý zvuk v co nejvyšší kvalitě prostřednictvím IP protokolu. Přenos zvuku je klasická telekomunikační úloha, jejíž počátky sahají do druhé poloviny 19. století. Postupem času, jak se zlepšovaly technické možnosti a vědecké znalosti, se zlepšovaly i parametry a možnosti telefonie. S rozvojem počítačů a sítí, které je propojují, se telefonie stále častěji realizuje za pomoci těchto technologií. Analogové ústředny jsou nahrazovány digitálními a kde je to možné, tam se využívá paketového principu přenosu signálu. Čím kvalitnější zvukový signál, tím více dat je třeba přenést. S rostoucím množstvím dat se také zvyšují nároky na přenosové pásmo. Toho je však obecně spíše nedostatek, a proto je nutná úspora dat pomocí různých kompresních technik.

Zařízení je postaveno na počítačové mikroplatformě umožňující vzdálený poslech přes protokoly SIP a HTTP. SIP protokol je řídicí protokol internetové telefonie. HTTP je internetový protokol přenášející dokumenty ve formátu HTML.

V první kapitole této práce jsou popsány principy technik internetové telefonie. Jak se zpracovává zvukový signál, jak se s ním dále pracuje a přenáší se sítí. Jakých internetových protokolů se používá k přenosu užitečných dat a signalizace a jakým způsobem je IP síť rozvrstvena a řízena.

Druhá kapitola se zabývá možnostmi hardwarového řešení, především výběrem a popisem parametrů mikroplatformy, na které je celé zařízení postaveno. Právě schopnosti mikroplatformy jsou rozhodující při realizaci zařízení s pomocí jednotlivých aplikací. Nedílnou součástí je samozřejmě výběr vhodné zvukové karty a mikrofonu.

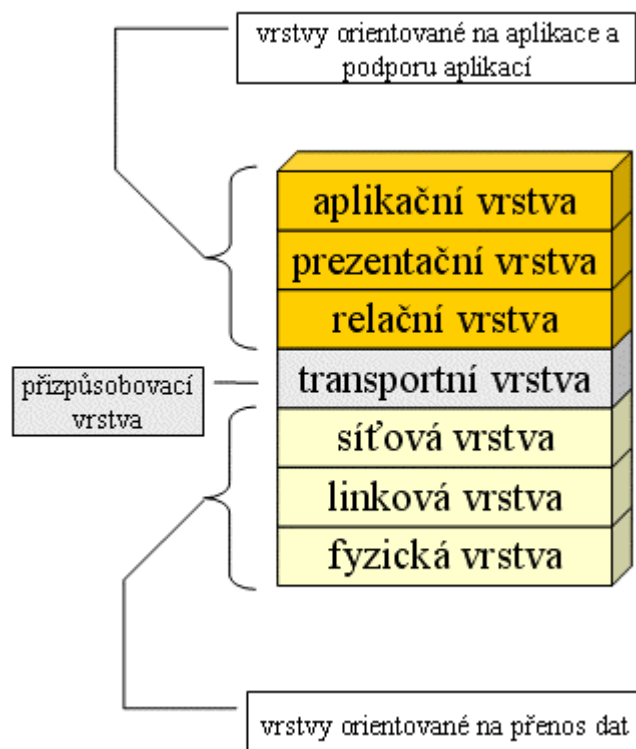
Třetí kapitola se věnuje samotné realizaci zařízení, volbě vhodné linuxové distribuce a aplikací, jejich instalacím, úpravám, nastavením a řešení vzniklých problémů. Dále se zabývá vytvořením webových stránek pro poslech pomocí HTML5, konfigurací VoIPové ústředny, nastavením možností autentizace uživatelů a vytvořením podmínek pro poslech v co nejvyšší kvalitě.

## 2. Přenos zvuku prostřednictvím IP sítě

Základním stavebním kamenem IP sítě je Internet Protocol, který pracuje na síťové vrstvě definované Referenčním modelem ISO/OSI (jehož každá vrstva vykonává přesně definovanou funkci, poskytuje služby sousední vyšší vrstvě a využívá služeb sousední nižší vrstvy). Internet Protocol poskytuje datagramovou službu dalším protokolům. Datagram je základní jednotka putující počítačovou sítí a přepravující data (nejčastěji paket). Datagram obsahuje hlavičku, jejíž částmi jsou cíl a odesílatel (pomocí IP adresy), ale nezaručuje správné doručení (zda datagramy dorazí ve správném pořadí, zda nedorazí víckrát či zda vůbec dorazí). O to se starají jiné protokoly.

### 2.1 Struktura počítačové sítě

V roce 1984 byl mezinárodní organizací pro normalizaci ISO přijata norma 7498 o standardu počítačových sítích OSI (Open System Interconnection), jejíž součástí je i Referenční model ISO/OSI.



Obr.1: Referenční model ISO/OSI (převzato z [13]).

**Fyzická vrstva** – má na starost přímé ovládání (navazování, konfiguraci a ukončování) spojení s fyzickým přenosovým médiem (sériová linka, Ethernet, huby, opakovače apod.). Data jsou prezentována jednotlivými bity.

**Linková (spojová) vrstva** – její hlavní činností je poskytnutí spojení mezi dvěma sousedními systémy a práce s daty z Fyzické vrstvy. Data uspořádává do rámců (které opatřuje MAC adresami), stará se o jejich seřazení, oznamuje když je nelze opravit (kontrolní součty, parita atd.). Ethernet je realizován právně na této a fyzické vrstvě.

**Síťová vrstva** – má na starost směrování a síťové adresování. Spojuje dva nesousední systémy. K přenosu dat používá pakety, které směřuje od odesílatele k cíli. Právě na této vrstvě pracuje Internet Protocol (v dnešní době se přechází z verze 4 na verzi 6).

**Transportní vrstva** – jejími nejpoužívanějšími protokoly jsou TCP, UDP a SCTP. Nestará se o směrování, ale o spolehlivý přenos dat a umožňuje adresování jednotlivých aplikací na konkrétní porty v TCP/IP (rodina protokolů pracujících na komunikaci v počítačové síti).

Následující tři vrstvy jsou v architektuře TCP/IP sloučeny do jedné pod názvem Aplikační.

**Relační vrstva** – vytváří, ovládá a ukončuje spojení. Pro určení správného pořadí paketů jim připojuje synchronizační značky. SSL protokol sice pracuje mezi touto a Transportní vrstvou, ale řadí se do této vrstvy.

**Prezentační vrstva** – tato vrstva transformuje data do takového tvaru, aby jim rozuměly aplikace, které s nimi pracují.

**Aplikační vrstva** – nejvyšší vrstva jejíž činností je poskytnutí služeb IP síť konkrétním aplikacím. Na této vrstvě pracují HTTP, SIP, SSH a další.

## 2.2 VoIP

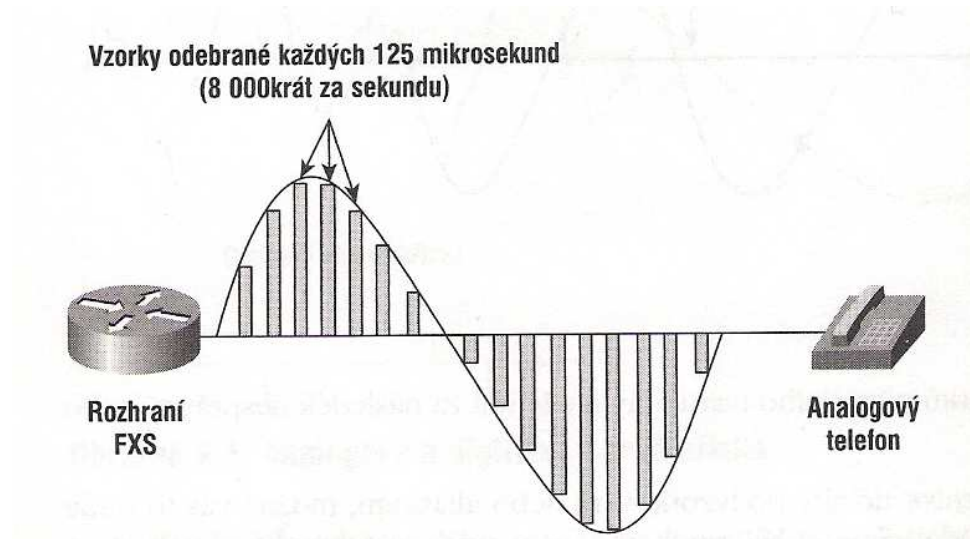
Voice over IP (VoIP) je technologie paketového přenosu digitalizovaného hlasového signálu datovou sítí (Internet). Využívá protokolu H.323 nebo SIP (Session Initiation Protocol). Přenos hlasových dat je proveden pomocí protokolu RTP (Real-time Transport Protocol). Pakety RTP jsou přenášeny uvnitř datagramů UDP (User Datagram Protocol) protokolu, který je vhodný pro Real-time využití (jen výjimečně je použit TCP). Naproti tomu přenos signalizace je uskutečněn protokolem TCP (Transmission Control Protocol).

### 2.2.1 Zpracování hlasového signálu

Datovými sítěmi se data přenáší v digitální formě. V rámci digitalizace analogového hlasového signálu dojde nejprve k vzorkování a kvantizaci. Digitalizovaný signál pak z důvodu uspořádanosti objemu přenášených dat zkomprimujeme, čímž snížíme požadovanou šířku pásma.

### 2.2.1.1 Vzorkování

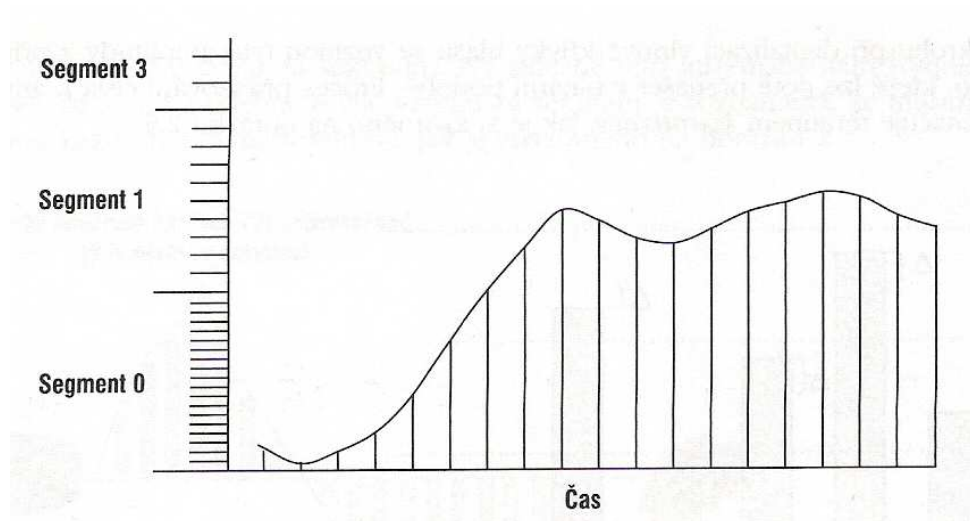
Není nutné přenést celý frekvenční rozsah řeči. Devadesát procent hlasové informace potřebných pro přenos čisté a srozumitelné řeči je obsaženo v 0-4kHz. Tím je podle Nyquistovy věty (Shannonův teorém – ke správnému vzorkování postačí dvojnásobná frekvence než je frekvence vzorkovaného signálu) definována vzorkovací frekvence telefonního kanálu 8kHz.



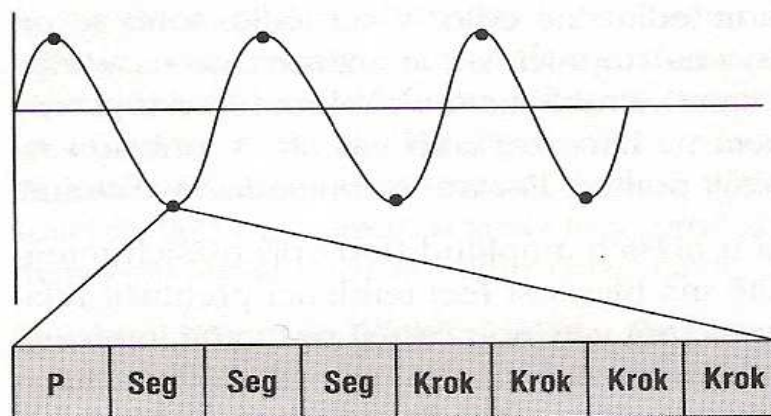
Obr.2: Vzorkování (převzato z [2])

### 2.2.1.2 Kvantování

V dalším kroku dochází k přiřazení osmibitového čísla amplitudě vzorků. Používá se logaritmická stupnice, neboť v lidské řeči jsou více obsaženy nižší frekvence.



Obr.3: Kvantování (převzato z [2])



1 bit polarity  
 3 bity segmentu  
 4 bity kroku  
 -----  
 8 bitů celkem

Obr.4: Význam jednotlivých bitů (převzato z [2])

### 2.2.1.3 Kódování

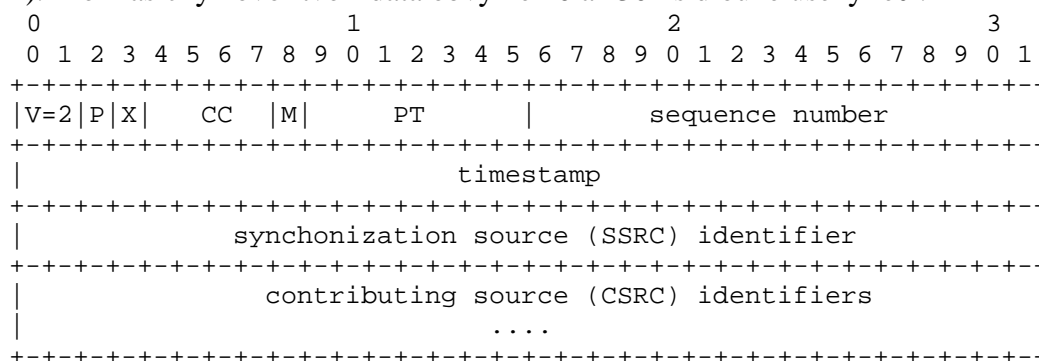
Předcházející postup popisuje modulaci PCM (Pulse Code Modulation) a používá ji kodek G.711. V rámci tohoto kodeku nedochází k žádné kompresi přenášených dat a je vyžadována šířka přenosového pásma 64kb/s (8kHz\*8bit).

Další velmi používaný zástupce kodeků je G.729 využívající CS-ACELP (Conjugate Structure Algebraic Code Excited Linear Prediction). Dynamicky vytváří knihu kódů a pokud je nějaký vzorek obsažen v této knize, pošle pouze adresu tohoto vzorku v kódové knize. Dojde tím k uspoření dat potřebných k přenosu informace o hlasovém vzorku. Tento kodek se dočkal i rozšíření, jako je například využití VAD (Voice Activity Detection), které například místo přenosu plnohodnotného ticha pošle pouze informaci o tichu. Tento kodek potřebuje šířku pásma 8kb/s.

## 2.2.2 Přenos dat

### 2.2.2.1 RTP

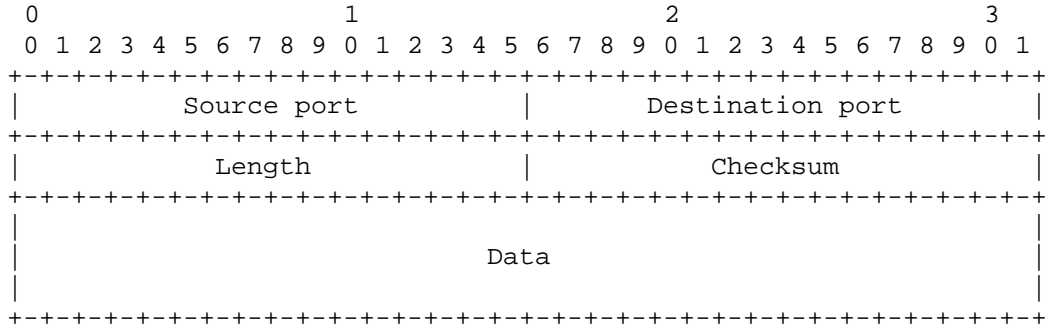
Hlavní činností tohoto protokolu jsou seřazení paketů (Sequence Number), které neumí protokol UDP, označení (Timestamp), díky kterému je rozpoznatelný rozptyl zpoždění mezi zpožděním očekávaným a skutečným (jitter), multiplexování a demultiplexování. Paket tohoto protokolu se skládá z bloku dat v RTP (20B až 160B velký) a hlavičky (12B). Pro klasický hovor tvoří data obvykle 20 až 30ms dlouhé úseky řeči.



Obr.5: Hlavička RTP

### 2.2.2.2 UDP

Je to nespojově orientovaný protokol transportní vrstvy, který nezaručuje doručení, doručení ve správném pořadí či jednonásobné doručení. Jeho prioritou je rychlost, a proto je také využíván k přenosu dat. Datagram UDP vznikne přidáním hlavičky před paket RTP. S tím, jak přidáváme další data, se samozřejmě zvyšuje i požadavek na šířku pásma zhruba o 26kb/s (pro G.711 i G.729).



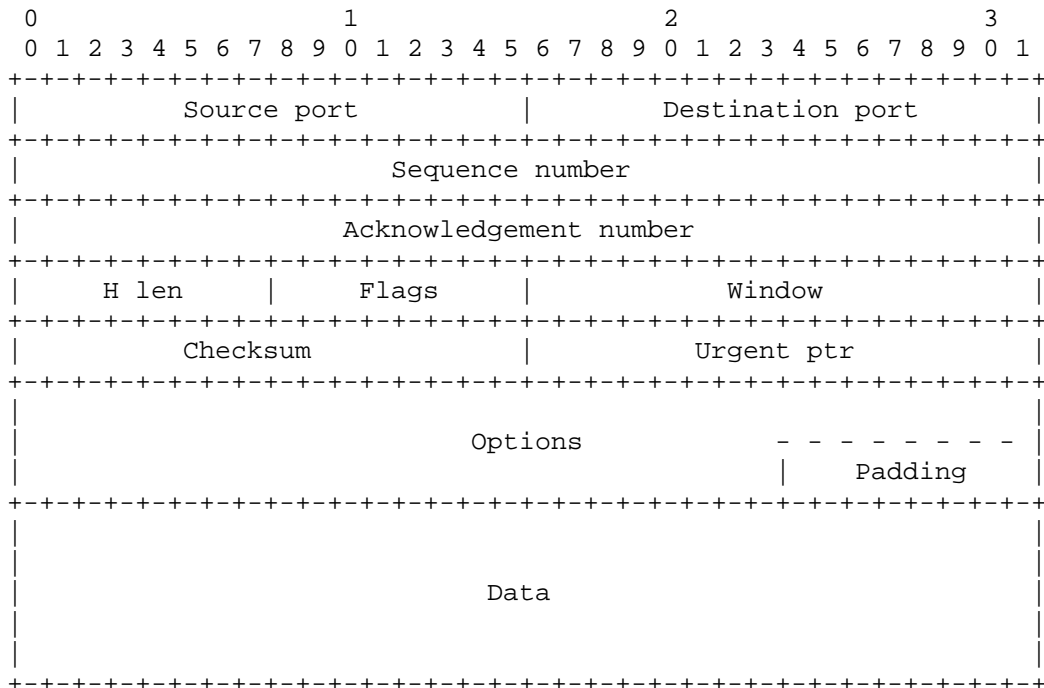
Obr.6: Hlavička UDP

### 2.2.3 Přenos signalizace

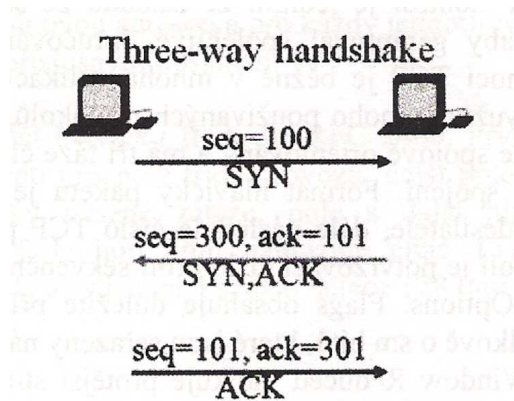
Neméně důležitou částí přenosu je samozřejmě i signalizace. Ta slouží k vytvoření spojení, správnému průběhu i k jeho korektnímu zakončení. V rámci IP telefonie se jako první rozšířil standard H.323. V dnešní době je rozšířenější protokol SIP, který vychází z protokolů HTTP (Hypertext Transfer Protocol) a SMTP (Simple Mail Transfer Protocol) a je mnohem jednodušší.

#### 2.2.3.1 TCP

Je to spojově orientovaný protokol transportní vrstvy, který garantuje spolehlivé doručení ve správném pořadí. Jeho činnost se skládá ze tří částí (navázání spojení – „3-way handshake“, přenos a ukončení).



Obr.7: Hlavička TCP



Obr.8: Navázání TCP spojení (převzato z [1])

## 2.2.4 Implementace VoIP

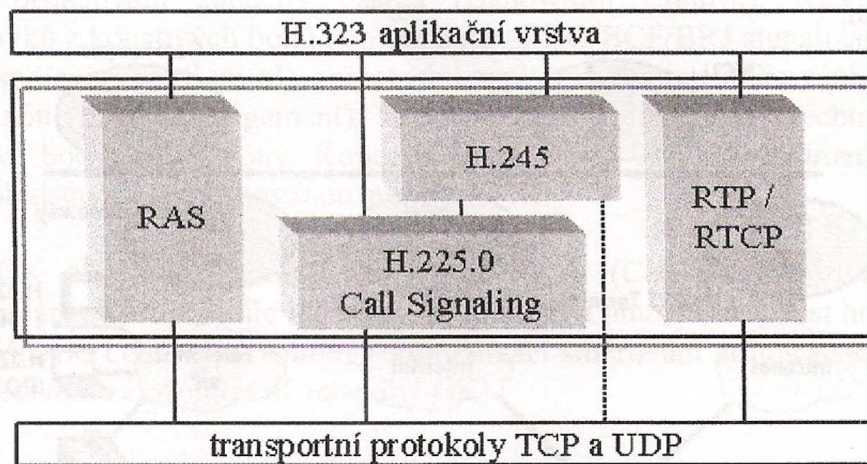
Kromě firemních implementací, jako jsou Skinny (Cisco) nebo HFA (Siemens), existují řešení H.323 a SIP, lišící se signalizací. Obě řešení umí přenést i videohovor.

### 2.2.4.1 H.323

Tento protokol byl uveden v roce 1996 a vychází z potřeb klasické telefonie (“chytrá“ ústředna, “hloupé“ terminály). To ho sice činí nejpropracovanějším, ale zároveň zbytečně složitým. Jak je uvedeno výše, data jsou přenášena pomocí protokolu RTP. Zde se v audio terminálu vyrovnává zpoždění doručovaných paketů (potlačení jitteru zajistí kontinuální dekódování). Maximální zpoždění mezi odesílatelem a příjemcem je 150ms (podmínka vysoké kvality). Ve standardu H.323 jsou stavové a řídicí informace (rozšířené zprávy dohledu – ztráty a vyřazené pakety, shluky ztrát, metriky zpoždění atd.) přenášeny pomocí protokolu RTCP (Real Time Control Protocol). Signalizace je přenášena pomocí protokolu TCP (výjimkou je signalizace RAS).

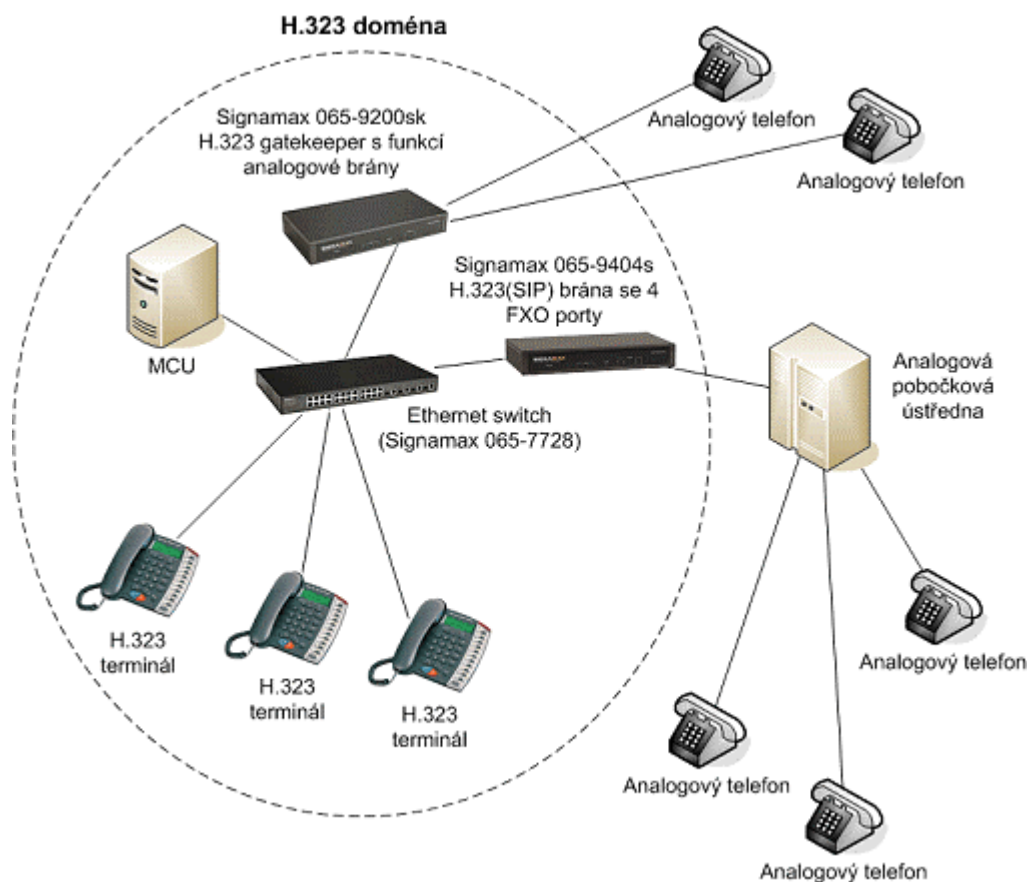
- RAS – řídí registraci, přístup a stav  
Registration, Admission and Status je protokol pro komunikaci s GK (Gatekeeper)
- Q.931 – řídí nastavení volání a jeho ukončení  
Call signaling je signalizace volání (SETUP, ALERTING, CONNECT atd.)
- H.245 – určí využití kanálu a jeho kapacitu  
Media control je protokol řízení médií (zabývá se kodeky, porty atd.)
- H.325 – zabezpečení a identifikace
- H.450 – doplňkové služby





Obr.9: Protokolový model H.323 (převzato z [1])

Struktura sítě v H.323 vychází z klasické telefonní sítě. Jejími součástmi jsou koncové body EP (endpoints), které jsou řízeny GK (Gatekeepers). Každý GK řídí zónu (doménu) s jemu přidělenými koncovými body, ale v případě potřeby může jako záloha převzít řízení dalších koncových bodů, spravovaných jiným GK. Koncový bod může být jak TE (koncový terminál), tak multikonferenční jednotka MCU (Multipoint Control Unit – řídí komunikaci mezi více koncovými body) nebo brána GW (Gateway – propojení s telefonní sítí).



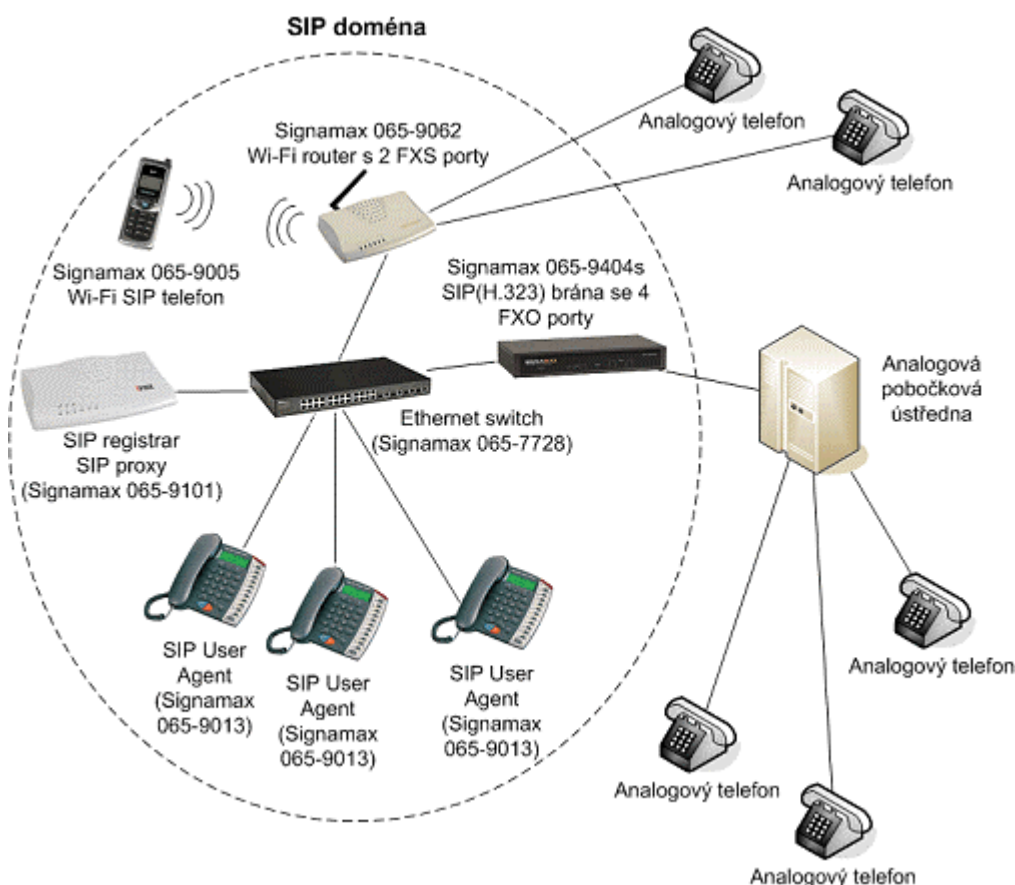
Obr.10: Struktura H.323 (převzato z [11])



V roce 2001 byl vytvořen open-source projekt s názvem GNU Gatekeeper, což je plně H.323 kompatibilní software. Je poskytován pro operační systémy Windows, Linux, MacOS, Solaris a FreeBSD. Dalším softwarem umožňujícím propojení s H.323 je Asterisk (ten je možné propojit s GNU GK).

#### 2.2.4.2 SIP

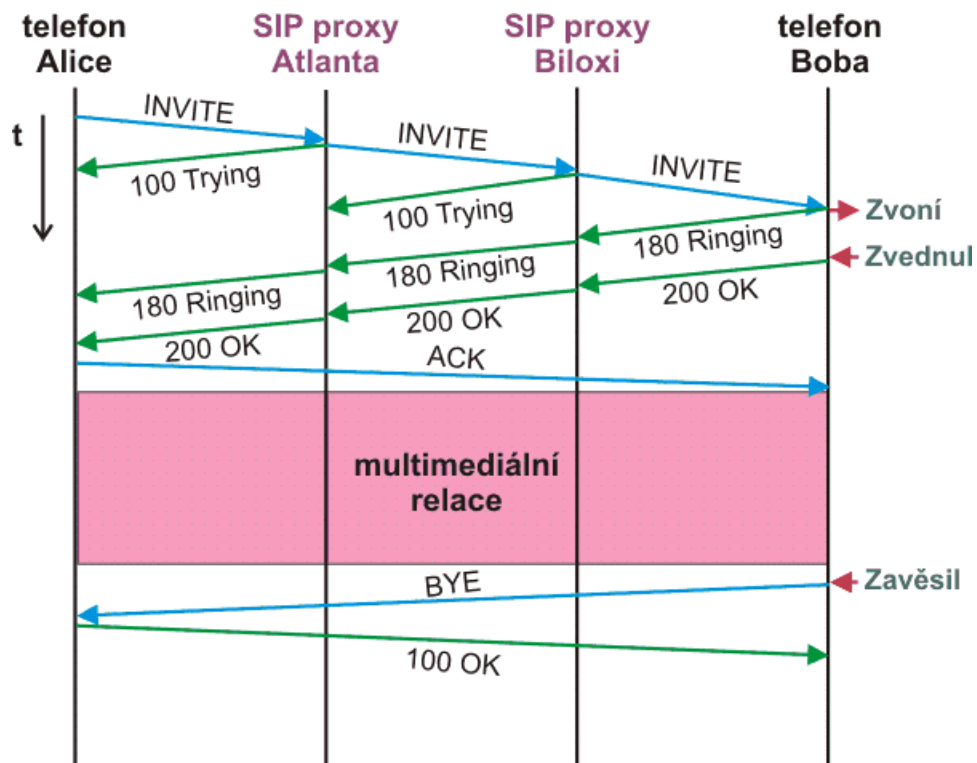
Session Initiation Protocol byl vyvíjen od roku 1996 a roku 2002 byl standardizován jako RFC 3261. Je to řídicí protokol pracující na aplikační vrstvě. Jeho výhodami jsou flexibilita, snadná implementace a rozšiřitelnost. Nevyužívá jen RTP/RTCP (resp. UDP), ale i SDP (Session Description Protocol – řídí správné přidělení typu média, transportního protokolu, typu kodeku či přenosové rychlosti). Dalším rozdílem oproti H.323 je jeho decentralizovanost, neboť veškerá logika je uložena v koncových zařízeních. Cenou za to je navýšení obsahu hlavičky, ale odměnou je vyšší účinnost sítí a možnost nových služeb. SIP je textově orientovaný podobně jako HTTP či SMTP (nejrozšířenější a nejpoužívanější protokoly Internetu). Podobně jako v H.323 spravuje Gatekeeper v jemu přidělené zóně koncové body, v SIP spravuje koncové body v přidělené entitě (doméně) SIP Proxy.



Obr.11: Struktura SIP (převzato z [11])

Tyto entity jsou identifikovány SIP URI (Uniform Resource Identifier). Ta se skládá ze dvou částí. User identifikuje uživatele a host určuje doménu (sip:user@host). SIP adresa je vázána k doméně a vyjádřena buď 32-bitově (IPv4) nebo 128-bitově (IPv6). DNS (Domaine Name System) přeloží doménová jména na IP adresy (nebo naopak). Koncové terminály UA (User Agents) mohou být jak SIP telefony tak i aplikace (např.

PSTN brány). Základní typy SIP zpráv jsou žádost (slouží k inicializaci) a odpověď (pokud má číselné označení jedná se o chybové hlášení).



Obr.12: Transakce protokolu SIP (převzato z [15])

- INVITE – žádost o spojení nebo změnu parametrů již probíhajícího
- 100 Trying – odpověď (v tomto případě to znamená, že krok probíhá v pořádku, ale ještě není ukončen)
- 180 Ringing – odpověď UA na začátek zpracování INVITE (např. začátek zvonění)
- 200 OK – odpověď na úspěšně ukončený krok (např. zvednutí sluchátka)
- ACK – odpověď na úspěšné sestavení spojení
- BYE – ukončení spojení
- CANCEL – zrušení spojení ještě před sestavením
- REGISTER – žádost o registraci (sváže IP adresu s portem) či odregistrování
- OPTIONS – podobné jako INVITE, ale není sestaveno spojení, jen žádá o informace

Jednotlivé UA mohou sice navázat komunikaci mezi sebou, ale většinou k tomu využijí SIP Proxy server. Tyto servery zajišťují směrování žádostí o spojení, účtování, autentizaci, registraci (pokud není implementován SIP Registrar Server) atd. Jsou buď stateless (bezstavové – pouze přeposílají zprávy, aniž by je uměly kontrolovat – jsou vhodné jako balanční) anebo statefull (s informací o stavech – vytvoří a uchovávají informace do ukončení transakce či dialogu). Statefull SIP Proxy server umí větvit (jedna zpráva je odeslána na vícekrát), kontrolovat a zachytit opakování zpráv, přesměrovat (pokud není implementován SIP Redirect server), účtovat atd. I zde lze použít softwarové řešení pomocí Asterisku.

## 2.2.5 SW klienti IP telefonie

Existuje mnoho desktopových programů umožňujících VoIP komunikaci. Jejich největší nevýhodou je potřeba doinstalování dalšího pluginu (rozšiřující software) jako je například Flash Player.

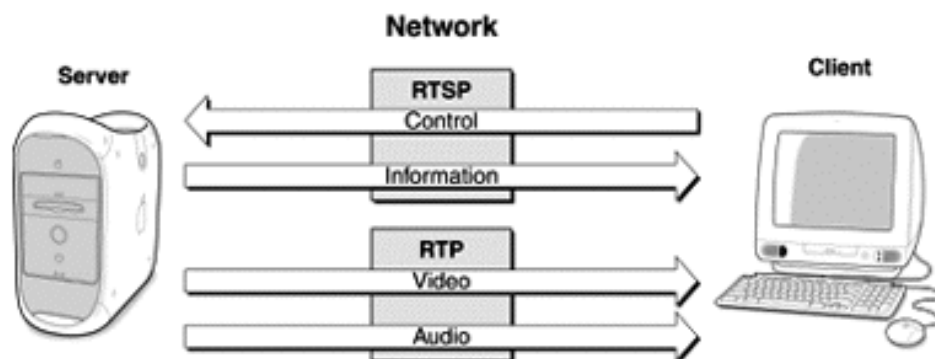
- Microsoft NetMeeting (H.323)
- Skype (použití je možné jen s připojením na Internet)
- IPphone (SIP)
- X-Lite (SIP)
- Tlen.pl
- Ekiga (H.323 i SIP)
- Empathy (SIP)
- linphone (SIP)
- SJphone (SIP)
- wxCommunicator (SIP)
- TeamSpeak
- Ventrilo
- Mumble

## 2.3 Streamování zvuku

Přenos zvuku ještě nutně nemusí znamenat, že se jedná o hovor. Poslouchat je možné také například internetové rádio.

### 2.3.1 Jednocestný stream

Pokud se jedná pouze o jednosměrný přenos zvuku a internetový prohlížeč či vhodný přehrávač, je možné přehrávat zvukový stream přímo z internetových stránek (zatím vyžaduje instalaci pluginu Flash Player, ale nově vznikající HTML5 to zvládá i bez něj) nebo zadáním url adresy streamu přímo v přehrávači. Zdrojem takového streamu je buď uložená audio nahrávka nebo zvuková karta. Stream klient (např. Ices2 či DarkIce) zpracovává data do nastavených parametrů jako jsou šířka přenosového pásma, vzorkovací frekvence či kodek (mp3, Ogg Vorbis, AAC, atd.) a odesílá je do stream serveru (SHOUTcast, IceCast). Stream server na portu 8000 (může být jiný) umožní aplikacím (jiným stream klientům) připojení ke konkrétnímu streamu. Přenos dat je uskutečněn protokolem RTP (RTCP) a signalizace protokolem RTSP (Real Time Streaming Protocol) nebo protokolem MMS (Microsoft Media Server), který zvládne oboje. Případně protokolem HTTP Live Streaming od společnosti Apple.

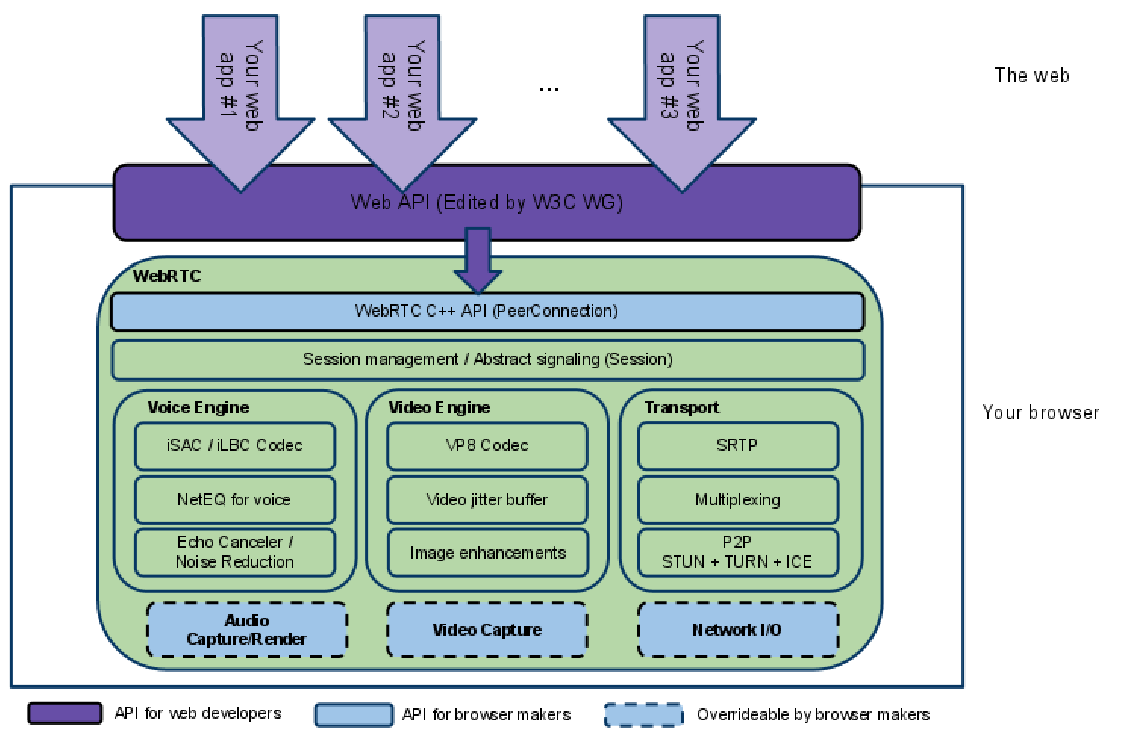


Obr.13: Komunikace mezi klientem a serverem (převzato z [14]).

### 2.3.2 Dvoucestný stream - WebRTC

Nevýhoda řešení komunikace mezi účastníky pomocí SW klientů (nutnost jejich instalace a registrace do VoIPové ústředny) vedla ke snaze oprostít se od nich a začlenit komunikační služby přímo do webového prohlížeče (Gogole Chrome, Windows Internet Explorer, Mozilla Firefox, Opera, Safari atd.) v rámci HTML5, JavaScript či CSS (Cascading Style Sheet). W3C (World Wide Web Consortium, které vytváří HTML i CSS) začalo v roce 2011 pracovat na novém standardu WebRTC (Web Real-Time Communication), který umožní webovým prohlížečům používat aplikace hlasového volání (také videohovory a sdílení dat) bez dalších pluginů. Většina velkých webových prohlížečů již zavádí WebRTC do svých nejnovějších verzí. Neboť výhoda toho, že uživatelé budou moci tyto služby použít na jakémkoliv počítači pouze spuštěním prohlížeče (není nutné mít administrátorská práva k instalaci nepříliš bezpečných pluginů, stačí jen připojení k Internetu), je nesporná. Další potenciál uplatnění je v dnes velmi používaných sociálních sítích (Facebook, Twitter, MySpace, Spolužáci atd.), jejichž autoři budou moci tyto služby nabídnout přímo v menu.

První SIP klient založený na WebRTC vytvořila společnost Doubango Telecom, použili k tomu pouze HTML5 a JavaScript (to svědčí o nadčasovosti SIPu a jeho bezproblémové implementaci).



Obr.14: Struktura WebRTC (převzato z [15]).

### 2.3.3 Zpracování hlasového signálu

Při streamování se pracuje s hlasovým signálem jak je tomu běžné u hlasových aplikací. Rozdíl je tom, že se používá i pro přenos (sdílení) hudby a videa, a je tudíž kladen důraz na co nejvyšší kvalitu. Signál je digitalizován a poté komprimován pomocí kodeku (PCMA/PCMU, Telephone Event, iSAC, iLBC či Opus – je schopen vyšších vzorkovacích frekvencí (48kHz) a je přednostně nabízen v případě potřeby vzorkovací frekvence vyšší než 8kHz). Další vlastností nutnou pro správné fungování komunikace

je potlačení echa mezi mikrofonem a reproduktorem (týká se například zabudovaných komponent u notebooků).

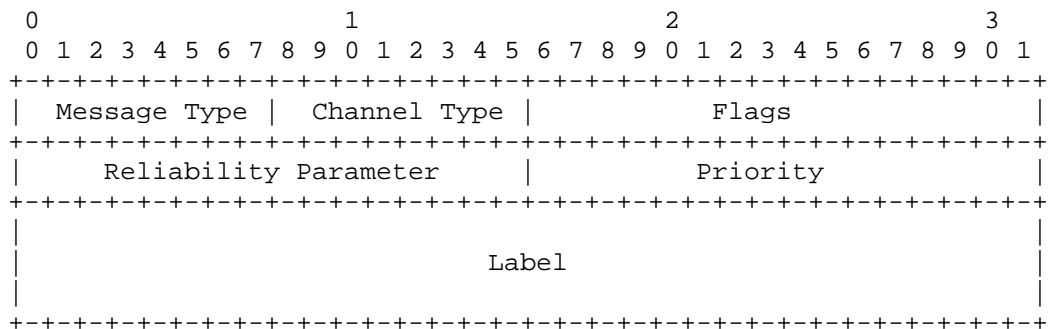
### 2.3.4 Přenos dat

Pro přenos dat streamem je použit osvědčený protokol RTP (řízen RTCP), k jehož přenosu je použit protokol UDP. Vzhledem k tomu, že se na trase spojení dvou prohlížečů budou vyskytovat NAT (Network Address Translation) i Firewally, RTP a RTCP se v rámci WebRTC sloučí do jednoho toku (úspora času).

### 2.3.5 Přenos signalizace

Jak je zmíněno výše k řízení RTP je použit RTCP. I zde platí, že signalizace slouží k vytvoření spojení, správnému průběhu i k jeho korektnímu zakončení. Navázání spojení probíhá „3-way handshakem“ a podílí se na něm SCTP (Stream Control Transmission Protocol – Protokol transportní vrstvy, zvládá vytvořit více paralelních toků a ty používá (v základním nastavení 16)), DTLS (Datagram Transport Layer Security – řídí práci s pakety) a SDP.

Nejprve je poslána žádost (DATA\_CHANNEL\_OPEN\_REQUEST), odpověď na ni (DATA\_CHANNEL\_OPEN\_RESPONSE) a na závěr potvrzení o vytvoření (DATA\_CHANNEL\_ACK).



Obr.15: Žádost DATA\_CHANNEL\_OPEN\_REQUEST

V průběhu spojení se kontroluje kvalita signálu, požadavky na změnu (např. navýšení objemu dat při zvýšení hlasitosti nebo úsporu když je ticho) ztrátovost či zpoždění jednotlivých toků (v případě zhoršení kvality toku se tento tok vyřadí) atd.

## 3. Hardwarové řešení

Celý systém se dá rozdělit do dvou částí. Základní hardware, na kterém bude nainstalován veškerý pomocný software (servery, VoIP ústředna, přehrávač, streamovací klient atd.) a hardware zpracovávající zvuk.

### 3.1 Základní hardware

Podle specifikace činnosti celého zařízení bylo nutné, aby základní hardware měl dostatečnou výpočetní kapacitu (vhodný typ procesoru, dostatečný výkon procesoru a operační paměti), odpovídající vstupní/výstupní periférie a co nejširší podporu softwarových platforem a aplikací pro řešení jednotlivých úkonů. Hned v úvodu nás zaujala novinka na trhu Raspberry Pi a rozhodli se na ní realizovat tuto práci. Raspberry Pi samozřejmě není jediným takovým zařízením na trhu a proto později uvedu několik dalších možných hardwarových platforem.



Obr.16: Raspberry Pi model B (převzato z [12]).

#### 3.1.1 Procesor a RAM

Základním prvkem Raspberry Pi je procesor ARM1176JZF-S (Advanced RISC Machine) taktovaný na 700 Mhz. RISC (Reduced Instruction Set Computing) architektura procesoru je založena na jednoduché, vysoce optimalizované sadě instrukcí jednotné délky jednoho cyklu. Takovýto procesor využívá mikroinstrukce (které jsou realizovány hardwarově přímo na procesoru, což zvyšuje rychlost jejich provádění) a pipelining (řetězení instrukcí). Další výhodou procesoru ARM je nízká spotřeba energie

(napájení 5V/1A by bylo možné realizovat i pomocí baterií). Vzhledem k použití nevyužijeme grafický procesor VideoCore IV.

První model Raspberry Pi (tedy model A) pracoval pouze s 256MB, ale model B (v prodeji od října 2012) má již 512MB RAM.

### 3.1.2 Vstupní/výstupní periferie

Model B obsahuje dva USB 2.0 porty pro připojení dalších zařízení (klávesnice, myš atd.). Obrazové výstupy (HDMI, Composite RCA a DSI), zvukové výstupy (HDMI, 3,5jack). Ethernetový adaptér 10/100 s konektorem RJ45 a slot pro SD kartu, ze které se bude bootovat. Dále pak 8xGPIO, UART, I2C a SPI sběrnici.

### 3.1.3 Operační systém

Raspberry Pi je možné používat s předpřipraveným obrazem linuxové distribuce (Raspbian, Soft-float Debian, Arch linux ARM nebo RISC OS). Zvláště Arch linux ARM nabízí širokou nabídku softwarových balíčků v repozitářích.

### 3.1.4 Další možnosti řešení

#### TC-261MXK

Tento výrobek firmy XtendLan používá procesor Vortex86/1GHz s 512MB RAM a VGA paměť 32MB. Má 3xUSB 2.0, BootRom PXE/RPL, slot pro SD kartu (ze které je možné bootovat), audio výstup a mikrofonní vstup (2xjack3,5), Ethernet 10/100 (RJ-45). Napájení řešeno adaptérem 5V maximálně 2A. Podporuje Windows, DOS i Linux.



Obr.17: Čelní panel TC-261MXK (převzato z [16]).

#### ODROID-U2

Korejský výrobce použil do své mikroplatformy procesor ARM Exino4412 Prime 1,7GHz s 2GB RAM. Má slot na microSD kartu, 3x USB 2.0, HDMI, audio výstup jack 3,5 a audiovstup pro digitální mikrofon, Ethernet 10/100 (RJ-45). Napájen je také adaptérem 5V/2A. Pracuje se systémy Android a Ubuntu.

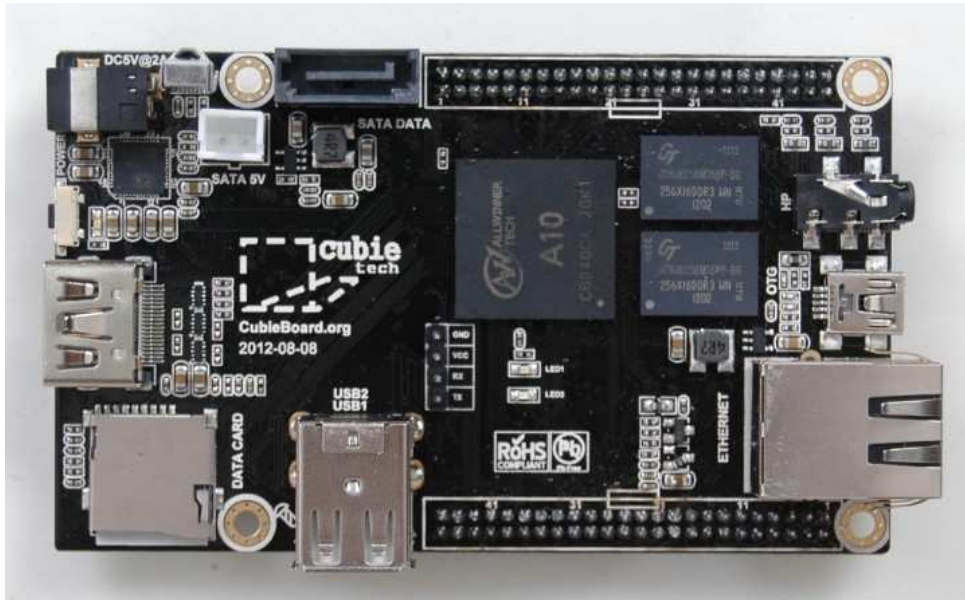




Obr.18: ODROID-U2 (převzato z [17]).

### Cubieboard

Toto miniPC čínské společnosti Cubietech je postaveno na 1GHz procesoru ARM cortex-A8 s až 1GB RAM. HDMI, 2x USB 2.0, slot na mikroSD kartu, SATA konektor, Ethernet 10/100 (RJ-45). Pracuje se systémy Android, Ubuntu a dalšími linuxovými distribucemi.



Obr.19: Cubieboard (převzato z [18]).

### KTT20/pITX

Firma Kontron svou mikroplatformu postavila na základě ARM. Svůj potenciál má tato platforma díky NVIDIA Tegra 2 DC procesoru se dvěma jádry hlavně v graficky náročných aplikacích. DVI, 2x USB 2.0, RS232, Ethernet 10/100 (RJ-45), napájení 5V/1A. Pracuje s Aneroid BSP, Linux BSP, Windows Embedded Compact 7 BSP.





Obr.20: KTT20/pITX (převzato z [19]).

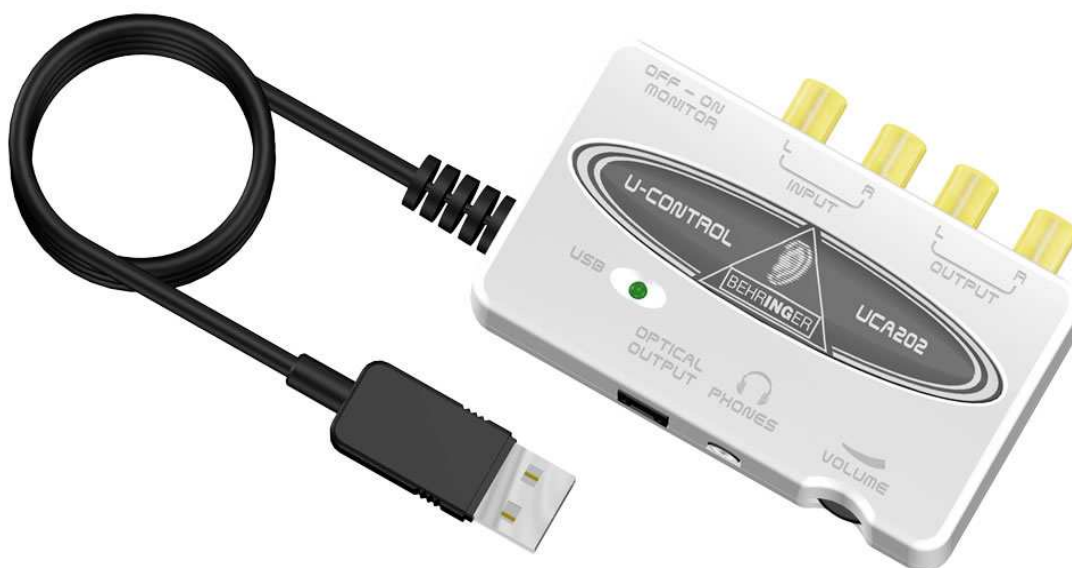
## 3.2 Hardware zpracovávající zvuk

Vzhledem k tomu, že zařízení by mělo být schopno snímat a přenášet zvuk v Hi-Fi kvalitě, tedy ve věrné kvalitě (nerozeznatelné od přímé řeči), je třeba použít externí USB zvukovou kartu. Zařízení integrovaná přímo na základní desku jsou nedostačující, neboť jsou navrhována především pro kvalitní reprodukci zvuku. Mikrofonní vstup v případě Raspberry Pi zcela chybí.

### 3.2.1 USB zvuková karta

Na trhu je velké množství externích zvukových karet, takže je vhodné stanovit minimální vyžadované vlastnosti této karty. Jak již bylo uvedeno, lidský sluch je schopen zachytit frekvence přibližně od 20Hz do 20kHz. Dle Nyquistovy věty je tedy nutné, aby zvuková karta pracovala s vzorkovací frekvencí alespoň 40kHz. Dalším kritériem je spotřeba energie, neboť Raspberry Pi je schopna do svých USB portů dodat pouze 100mA/5V maximálně, což je daň za nízkou spotřebu Raspberry Pi. Posledním kritériem, které je třeba vzít v potaz je, aby tato karta nevyžadovala žádné (plug-in) nebo alespoň co nejmenší nároky na instalaci (a existovaly ovladače pro linux).

Tyto vlastnosti splňuje například karta U-CONTROL UCA202 od německé společnosti Behringer zabývající se audiotechnikou. Tato externí zvuková karta pracuje s vzorkovací frekvencí až 48kHz a používá 16-ti bitové kvantování.



Obr.21: Behringer U-CONTROL UCA202 (převzato z [20]).

### 3.2.2 Mixážní pult/předzesilovač

Výše uvedená zvuková karta má dva RCA monofonní vstupy pro mikrofony. Jejich vstupní impedance je  $27k\Omega$ . Tato karta bohužel nemá zabudovaný předzesilovač a nějaký (případně mixážní pult ho obsahující) by musel být použit. Zvuková karta U-PHONO jej má, proto je lepším řešením. Použijeme-li mikrofon nevyžadující externí napájení, nebudeme potřebovat ani fantomový napaječ či mixážní pult.

### 3.2.3 Mikrofon

Mikrofon slouží k převodu akustického signálu na elektrický. Základním principem je snímání prohnutí membrány, na kterou dopadá akustická vlna. Podle principu snímání prohnutí membrány se mikrofony dělí na uhlíkové, piezoelektrické, dynamické, páskové, kondenzátorové a elektretové.

#### 3.2.3.1 Uhlíkový

Tento typ patří k nejstarším používaným mikrofونům. Tlak akustické vlny na membráně tlačí na zrnka uhlíku. Stlačením dochází ke změně elektrického odporu. Jeho kmitočtový rozsah je sice jen od 200 do 3400Hz, ale to přesně dostačuje běžnému využití v telefonii. Navíc je levný a snad vyrobitelný.

#### 3.2.3.2 Piezoelektrický

V tomto případě může být akustický tlak přenášen přímo na krystal Siegenntovy soli, jehož deformací dochází ke vzniku elektrického náboje. Jsou poměrně nekvalitní a využívají se jen jako kontaktní snímače akustických nástrojů.

#### 3.2.3.3 Dynamický

Cívka připevněná na membránu se pohybuje v magnetickém poli. Elektromagnetickou indukci vznikne elektrický proud odpovídající pohybu membrány. Jsou vhodné především pro hlasité zvuky. Permanentní magnet navíc nevyžaduje napájení.

### 3.2.3.4 Páskový

Jedná se o typ dynamického mikrofону, který místo cívky používá zvlněný pásek, který slouží jako membrána.

### 3.2.3.5 Kondenzátorový

Membrána je pohyblivou elektrodou kondenzátoru, jehož změna kapacity se projeví buď změnou napětí na připojeném velmi měkkém zdroji napětí nebo změnou kapacity rozladuje vysokofrekvenční oscilátor, jehož je součástí (méně běžné). Tento mikrofón má vysokou citlivost a je používán v profesionálních aplikacích a měřeních. Nevýhodou je nutné napájení.

### 3.2.3.6 Elektretový

Jedná se o typ kondenzátorového mikrofónu, který má jednu elektrodu tvořenou vrstvou elektretu (nevodivý elektricky permanentně nabitý materiál). Takový mikrofón vyžaduje pouze napájení vnitřního zesilovače, což je nejčastěji baterie.

Z výše uvedených mikrofónů vychází nejlépe elektretový mikrofón. Jeho kvalita je dostačující a nepotřebnost fantomového napájení z něj dělá nejpoužívanější mikrofón v řečových aplikacích (mobilní telefony, diktafony). Následující stereo mikrofón SBCME570 společnosti Philips má frekvenční rozsah 50-18000Hz, 600 $\Omega$  vstupní impedanci, všesměrovou charakteristiku a citlivost -45dB +/-3dB.



Obr. 22: Stereo elektretový mikrofón SBCME570 (převzato z [21]).

## 4. Realizace

Samotná realizace zařízení pro vzdálený poslech se bude skládat z instalace linuxové distribuce a vhodných aplikací, které budou zpracovávat jednotlivé úkony, na mikroplatformu Raspberry Pi. Půjde především o softwarovou pobočkovou ústřednu Asterisk, stream klienta, stream server a webový server.

### 4.1 Raspbian

Výběr z uvedených čtyř distribucí určených pro Raspberry Pi se z důvodu co nejširší nabídky balíčků v repozitářích (databázích) dá omezit na Arch linux ARM a Raspbian “wheezy“. Soft-float Debian “wheezy“ není vhodný z důvodu pomalejších aritmetických operací s hodnotami v softwarovém řešení plovoucí řádové čárky. Po postupných pokusech s oběma distribucemi vyšla nakonec lépe distribuce Raspbian “wheezy“, která vychází z jedné z nejstarších distribucí linuxu – Debian.

#### 4.1.1 Instalace

Raspberry Pi používá jako svou paměť SD kartu. Instalace Raspbianu se tedy skládá ze stažení aktuálního obrazu z oficiálních stránek [23] nebo z nějakého dalšího zrcadla.

V dalším kroku je třeba zabalený 471MB velký soubor rozbalit a například pomocí Win32DiskImageru naložovat do zformátované (např. SDFormatter) SD karty.

#### 4.1.2 Konfigurace

Je-li možnost připojit Raspberry Pi k monitoru, můžou se následující kroky uskutečnit v menu během prvního spuštění s připojeným monitorem.

##### **Rozšíření SD karty**

Obraz Raspbianu je vytvořen pro 2GB SD kartu. Při použití větší (což by bylo vhodné vzhledem k potřebě doinstalace dalších aplikací) lze rozšířit obraz na celou kapacitu. Například pomocí GParted.

##### **SSH přístup**

Obraz Raspbianu má vytvořený SSH klíč a umožňuje okamžité vzdálené připojení a správu, například pomocí PuTTY. Nejprve je však třeba zjistit jakou síťovou adresu mu router přiřadil. Použít lze Nmap, pomocí kterého se zkontrolují otevřené porty na Raspberry Pi. SSH umožňuje autentizaci a šifrovaný přenos dat.

##### **Heslo**

Prvním bodem po přihlášení do Raspberry Pi je změna hesla. V obrazu Raspbianu je nastaveno uživatelské jméno – pi a heslo – raspberry. Nyní nastal čas říct si něco málo o linuxových příkazech.

- cd – přesun do adresáře
- ls – vypsání obsahu adresáře
- cat – vypíše obsah souboru
- sudo – umožní provést operace jako superuživatel
- mkdir – vytvoří nový adresář
- rm – odstraní adresář nebo soubor
- nano – otevře textový editor nano, ve kterém je možné vytvořit či upravit textový soubor

- reboot – provede znovunačtení
- shutdown –h now – řádně ukončí systém a vypne

Po přesunu do adresáře etc (cd /etc), ve kterém jsou konfigurační soubory, se heslo změní příkazem passwd.

### **Časové pásmo**

V souboru timezone je potřeba nastavit Europe/Praque.

Další nastavení už je nutné provést ručně v adresáři etc.

### **Nastavení statické síťové adresy**

Vzhledem k budoucí činnosti Raspberry Pi je nutné platformě nastavit jednu konkrétní adresu (například 192.168.0.4) a tu uvést ve všech aplikacích, které se k ní budou připojovat.

```
sudo nano /etc/network/interfaces
```

a upravení vypadá následovně

```
auto lo
iface lo inet loopback
iface eth0 inet static
address 192.168.0.4
netmask 255.255.255.0
gateway 192.168.0.1
```

### **Přejmenování**

Raspberry Pi je pojmenováno jako raspberrypi. Přejmenování se provádí v /etc/hostname a následně v /etc/hosts.

## **4.2 Vytvoření streamu**

Jak jsem již uvedl, první pokusy byly provedeny s distribucí Arch Linux ARM.

V prvním kroku je třeba se rozhodnout, zda stream server bude vytvořen pomocí IceCast či SHOUTcast. Protože je žádoucí, aby stream server uměl přenést co nejkvalitnější zvuk v co nejmenší šíři pásma, je lepší zvolit IceCast2. Navíc umí přenášet data nejen v mp3 audio formátu, ale i ve formátu Ogg Vorbis.

Dalším krokem je výběr streamovacího klienta, který přenáší signál mezi zvukovou kartou (přístup přímo k ALSA-Advanced Linux Sound Architecture) a stream serverem. Jako první volbou byl Ices2 od tvůrců IceCast2. V Arch linuxu se dají programové balíčky instalovat buď z oficiálního repozitáře schválených a ověřených balíčků (případ IceCast2) nebo z AUR (Arch Linux User-community Repository), do nějž uživatelé vkládají vytvořené balíčky a ostatní uživatelé pro ně hlasují a ty nejlepší jsou po nějaké době přesunuty mezi oficiální. Neoficiální balíček se musí nejprve stáhnout, rozbalit, vytvořit pomocí makepkg balíček a teprve pak pacmanem nainstalovat.

Hned v počátcích se ukázala nevýhoda v podobě neschopnosti pracovat s jiným než Ogg Vorbis audio formátem. Mnohem větší problém se však projevil během samotného provozu. Přestože Ices2 zabíral kolem 30% kapacity CPU, byl přenos signálu ve kvalitě 44100Hz/16bit/96kbps/mono z jakéhosi důvodu poškozován a přehrávání ve Winampu probíhalo tak, že stejný čas probíhal buffering a stejný čas se přehrával signál.

Nedocházelo však k jeho postupnému přehrávání, ale přehrával se s dvojnásobnou rychlostí. Celkové zpoždění bylo stále stejné, zhruba 7 vteřin.

Bylo tedy třeba najít jiné řešení a jako první je volba jiného streamovacího klienta podobných parametrů. Tím je DarkIce. Bohužel, jak Ices2 tak DarkIce jsou neoficiální balíčky a nepodařilo se upravit balíček DarkIce tak, aby uměl pracovat se zadanými audio formáty. Bylo tedy nutné vyřešit, jak nainstalovat pozměněný balíček. Nalezený postup však byl pro balíček z repozitáře Raspbianu. Došlo tedy na změnu distribuce z Arch Linux ARM na Raspbian “wheezy“.

## 4.2.1 DarkIce

Následující postup je převzat z tutoriálu pro Raspberry Pi v [26].

### 4.2.1.1 Instalace

Nejprve je třeba uvést nový zdroj (repozitář) do /etc/apt/sources.list buď přímo nebo příkazem

```
sudo sh -c "echo 'deb-src http://mirrordirector.raspbian.org/raspbian/ wheezy main contrib non-free rpi' >> /etc/apt/sources.list"
```

a následně provést aktualizaci seznamu balíčků v repozitářích.

```
sudo apt-get update
```

Aby se DarkIce nainstaloval s patřičnými knihovny a doplňky je nutné je nainstalovat.

```
sudo apt-get --no-install-recommends install build-essential devscripts autotools-dev fakeroot dpkg-dev debhelper autotools-dev dh-make quilt ccache libsamplerate0-dev libpulse-dev libaudio-dev lame libjack-jackd2-dev libasound2-dev libtwolame-dev libfaad-dev libflac-dev libmp4v2-dev libshout3-dev libmp3lame-dev.
```

Do pracovního adresáře (například /src stáhnout zdrojový kód)

```
apt-get source darkice
```

a provést úpravu konfigurace kompilace v /src/darkice-1.0/debian/rules

```
#!/usr/bin/make -f
```

```
%:
```

```
dh $@
```

```
.PHONY: override_dh_auto_configure
```

```
override_dh_auto_configure:
```

```
ln -s /usr/share/misc/config.guess .
```

```
ln -s /usr/share/misc/config.sub .
```

```
dh_auto_configure -- --prefix=/usr --sysconfdir=/usr/share/doc/darkice/examples --with-vorbis-prefix=/usr/lib/arm-linux-gnueabi/hf/ --with-jack-prefix=/usr/lib/arm-linux-gnueabi/hf/ --with-alsa-prefix=/usr/lib/arm-linux-gnueabi/hf/ --with-faac-prefix=/usr/lib/arm-linux-gnueabi/hf/ --with-aacplus-prefix=/usr/lib/arm-linux-gnueabi/hf/ --with-samplerate-prefix=/usr/lib/arm-linux-gnueabi/hf/ --with-lame-prefix=/usr/lib/arm-linux-gnueabi/hf/ CFLAGS='-march=armv6 -mfpu=vfp -mfloat-abi=hard'.
```

Nyní je třeba změnit verzi balíčku, aby podporoval mp3 formát

```
debchange -v 1.0-999~mp3+1  
doplněním následujícího
```

```
darkice (1.0-999~mp3+1) UNRELEASED; urgency=low
```

```
* New build with mp3 support
```

Nyní už je možné balíček skutečně vytvořit

```
dpkg-buildpackage -rfakeroot -uc -b
```

a nainstalovat

```
sudo dpkg -i ../darkice_1.0-999~mp3+1_armhf.deb
```

#### 4.2.1.2 Konfigurace

DarkIce obsahuje příklad konfiguračního souboru darkice.cfg, který se zkopíruje

```
sudo cp /usr/share/doc/darkice/examples/darkice.cfg /etc/
```

a upraví podle vlastních požadavků.

```
[general]
```

```
duration      = 0  
bufferSecs    = 5  
reconnect     = yes
```

```
[input]
```

```
device        = hw:1,0  
sampleRate    = 44100  
bitsPerSample = 16  
channel       = 2          #stereo
```

```
[icecast2-0]
```

```
format        = mp3  
bitrateMode   = vbr  
quality       = 1.0  
bitrate       = 96  
server        = localhost  
port          = 8000  
password      = vlastní heslo, kterým se bude DarkIce prokazovat do IceCast2  
mnoutPoint    = High.mp3  
name          = HighQuality.mp3  
description   = Live stream 1  
url           = http://localhost  
genre         = thesis  
public        = no
```

```
[icecast2-1]
format      = mp3
bitrateMode = vbr
quality     = 1.0
sampleRate  = 8000
channel     = 1          #mono
bitrate     = 96
server      = localhost
port        = 8000
password    = vlastní heslo, kterým se bude DarkIce prokazovat do IceCast2
mnoutPoint  = High.mp3
name        = HighQuality.mp3
description = Live stream 1
url         = http://localhost
genre       = thesis
public      = no
```

Bohužel se ukázalo, že DarkIce zatěžuje CPU natolik, že Raspberry zvládne pouze jeden stereo (maximálně 44100Hz vzorkovací frekvence) a jeden mono (8kHz v.f.) stream. V této konfiguraci to je okolo 60%CPU. Protože neumí dešifrovat Ogg Vorbis a změnit vzorkovací frekvenci Ogg Vorbis streamu, je vhodné zvolit variantu s mp3 streamy. Ty jsou posílány ze zvukového zařízení hw:0,1 do streamovacího serveru IceCast2, jenž bude vytvořen v dalším kroku. Výpis dostupných zvukových zařízení umožňujících vstup zvukového signálu se získá pomocí příkazu `arecord -l`.

## 4.2.2 IceCast2

Použitím DarkIce místo Ices2 se odstranily problémy s přehráváním zvuku dvojnásobnou rychlostí, takže není nejmenšího důvodu, proč nepoužít jako streamovací server IceCast2.

### 4.2.2.1 Instalace

```
sudo apt-get install icecast2
```

V `/etc` se vytvořila složka `icecast2`, v níž se nachází konfigurační soubor `icecast.xml`.

### 4.2.2.2 Konfigurace

Konfigurace je velmi jednoduchá a spočívá v následujících úpravách. Více streamů je možné nastavit v sekci `limits`.

```
<sources>2</sources>
```

V sekci autentizace se nastaví heslo pro DarkIce.

```
<source-password>***</source-password>
```

Relay heslo sice nebude využito, protože sem nebudou přeposílány streamy z jiného stream serveru, ale je třeba ho vyplnit také.

```
<relay-password>***</relay-password>
```



Poslední heslo umožní se do stream serveru přihlásit jako administrátor. Je tak možné kontrolovat jednotlivé streamy a případně odpojit stream nebo posluchače.

```
<admin-password>***</admin-password>
```

Bohužel verze streamovacího serveru IceCast 2.2.6, která je nainstalována z repozitáře Raspbianu, nepodporuje SSL (Secure Sockets Layer). To je zakomponováno až do verze 2.3.2., takže správa stream serveru není šifrována.

### 4.3 SIP/Asterisk

Na úvod je potřeba říct, že Asterisk byl do Arch Linuxu instalován jako první software.

Prvotní myšlenka byla vytvořit v něm ALSA kanál s přiřazeným číslem. Chybou tohoto řešení je jednak to, že poslech je umožněn pouze jednomu posluchači, ale hlavně v tom, že ALSA kanál je automaticky nakonfigurován na kvalitu hovorového kanálu (8kHz vzorkování) a nezvládne se připojit ke zvukovému zařízení s jinou kvalitou.

Druhé řešení spočívalo, ve vytvoření konferenci, do které se připojil ALSA kanál a každý účastník, který se připojil do této konference, mohl poslouchat zvuk z mikrofону. Sám měl přitom zablokované vysílání. Toto řešení sice umožnilo poslech více posluchačům, ale stále zde byl problém s pouze hovorovou kvalitou zvuku.

Třetí variantou bylo použití MOH (Music On Hold), ve kterém bylo vytvořeno nikomu nepřirazené číslo a místo vyzváněcího tónu byl posluchači přehráván zvuk z mikrofону. Zvuk byl nahráván aplikací mpg123 do krátkého souboru a následně přehráván posluchači. Právě tento krátký soubor způsoboval nepříjemnou věc. Každému posluchači připojivšímu se v době, kdy nikdo neposlouchal, a tudíž se do souboru nenahrával aktuální zvuk, přehrál nejprve zvuk nahraný v závěru předchozího poslechu.

Čtvrté řešení spočívalo v tom, že zvuk bude přehráván jako MOH, který bude do Asterisku zasíláno streamovacím serverem. Toto řešení umožňuje poslech více uživatelům i volbu kvality zvuku volbou volaného čísla, které budou přehrávat různě kvalitní zvukový stream. Jak již bylo řečeno, během tvorby streamu došlo na změnu distribuce, takže pořadí výsledného postupu odpovídá pořadí zde uvedenému.

#### 4.3.1 Instalace

V repozitáři Raspberry Pi je připraven Asterisk-armhf 1:1.8.13.1~dfsg-3, který se nainstaluje následujícím příkazem.

```
sudo apt-get install asterisk
```

Během instalace se spustí průvodce instalací, ve kterém je třeba vyplnit ITU-T kód země (pro českou republiku 420).

#### 4.3.2 Konfigurace

Všechna nastavení, které se budou provádět, jsou v `/etc/asterisk`. Z kompletní instalace využijeme jen malou část.

- `asterisk.conf` – základní nastavení Asterisku, které se nebude měnit
- `modules.conf` – obsahuje jak se budou načítat jednotlivé moduly - neměnit

- sip.conf – obsahuje základní konfiguraci protokolu SIP včetně hesel uživatelů. Co zde bylo uvedeno je třeba vykomentujeme a vytvořit vlastní.

```
[general]
allowguests=yes           ;umožní anonymní volání

[authentication]

[adam]
type=friend               ;účastník smí volat i být volán
secret=***                ;heslo, kterým se prokazuje klient při přihlášení
userid=Adam <201>        ;zobrazení jména volajícího volanému
host=dynamic              ;účastník se může připojit z libovolné ip adresy
context=poslech          ;účastník bude v dialplanu uveden ve skupině poslech

[bohuslav]
type=friend
secret=***
userid=Bohuslav <202>
host=dynamic
context=poslech
```

- musiconhold.conf – zde se definuje odkud bude přicházet zvukový stream

```
[general]

[default]

[highmp3]
mode=custom               ;umožní použít aplikaci přehrávající stream
application=/usr/bin/mpg123 -q -s -m -r 8000 -b 1024 -@
http://192.168.0.4:8000/high.mp3
format=slin

[lowmp3]
mode=custom
application=/usr/bin/mpg123 -q -s -m -r 8000 -b 1024 -@
http://192.168.0.4:8000/low.mp3
format=slin
```

Aby bylo možné použít mpg123, je nutné ho nainstalovat. Hlavní důvod proč byl zvolen pro streamování formát mp3 je fakt, že ogg123 není bohužel v repozitáři Raspbianu “wheezy“ a Debian, ze kterého Raspbian obecně vychází, používá soft-float. Další nevýhodou je, že Asterisk, který je nainstalován, má nastaveno pro MOH vzorkovací frekvenci 8kHz. K použití s jinou vzorkovací frekvencí by bylo nutné kompilovat vlastní verzi Asterisku a patchovat toto nastavení.

- extensions.conf – obsahuje číslovací plán

```
[general]
static=yes
```

[globals]

[default]

exten => 231,1,Answer

exten => 231,2,MusicOnHold(lowmp3)

[poslech]

exten => 201,1,Dial(SIP/adam)

exten => 201,1,Dial(SIP/bohuslav)

exten => 232,1,Answer

exten => 232,2,MusicOnHold(highmp3)

Tím je umožněno neautorizovaným uživatelům volat na číslo 231 a poslech s nízkou kvalitou a autorizovaným i volání na číslo 232 s vysokou kvalitou zvuku. Všichni však budou poslouchat pouze monofonní zvuk, protože protokol SIP přenos stereo zvuku neumožňuje.

## 4.4 HTML5

Poměrně novou metodou přenosu streamu je použití jazyka HTML5, jehož součástí je modul audio, který umožňuje přehrávat zvuk v rámci webového prohlížeče bez použití doplňkového flash playeru.

### 4.4.1 HTTP server

Nejprve je nutné připravit webový server, na kterém budou umístěny webové stránky. Nejrozšířenější je Apache.

```
sudo apt-get install apache2 php5
```

### 4.4.2 Přehrávač

Na stránkách webového prohlížeče Opera je uvedený postup pro vytvoření přehrávače v HTML5, včetně hotových skriptů ke stažení. Použití je možné jeho upravenou verzí získanou z [35].

Soubory prohlížeče do Raspberry Pi lze přesunout například pomocí WinSCP, což je aplikace umožňující bezpečný přenos mezi lokálním a vzdáleným počítačem. SFTP (SSH File Transfer Protocol) je protokol, který pracuje ve spojení s protokolem SSH (přístup přes SSH již funguje), takže není třeba se o nic dalšího starat.

Aby přehrávač fungoval správně je nutné změnit v javascriptovém přehrávači player.js cestu na feed.json, v němž jsou uloženy cesty k příslušenství stanic (streamy a loga).

```
feedUrl: "http://192.168.0.4/player/radio/feed.json",
```

Dále se zde upraví i nastavení pro jQuery.

```
player.int("http:// 192.168.0.4/player/feed.json");
```

Bohužel jplayer umí pracovat buď s mp3 formátem nebo s Ogg Vorbis. Nastavit lze v player.js v setupPlayer:function() změnou na oggSupport: false, čímž se zablokuje formát Ogg Vorbis.

Vzhledem k tomu, že je vše řešeno pomocí mp3 formátu a prohlížeč Opera tento formát nepodporuje, je nutné použít jiný prohlížeč. Například Google Chrome podporuje oba uvedené formáty.

V dalším kroku se upraví samotný feed.json, ve kterém se změní odkazy streamů a log.

```
{
  "name" : "Low.mp3",
  "channel" : "Low.mp3",
  "logo" : "vlastní umístění",
  "middle" : {
    "type" : "middle",
    "mp3" : "http://192.168.0.4:8000/low.mp3"
  }
}
```



Obr.23: Webový HTML5 přehrávač živého streamu.

#### 4.4.3 Webové stránky

Teď už jen zbývá vytvořit webovou stránku (index.html v /var/www/), do které bude umístěn pro neautorizované posluchače odkaz na přehrávač monofonního zvuku s nízkou kvalitou.

```
<p href="umístění přehrávače1">zde</a>.
```

Autorizovaným posluchačům se po vyplnění hesla nabídne přehrávač obou streamů. Autorizace je řešena jednoduchým formulářem převzatým z [36], který směřuje uživatele na odkaz, jehož název odpovídá heslu.

```
<form name="formular" onsubmit="return false">
Heslo: <input type="password" size="10" name="heslo">
<input type="submit" value="Pokračovat" onclick="window.location.href =
'/player/radio2/' + document.formulare.heslo.value + '.html'">
</form>
```

## 5. Závěr

V rámci této práce bylo vytvořeno zařízení umožňující vzdálený poslech prostřednictvím IP protokolu. Zařízení je postaveno na mikroplatformě Raspberry Pi používající operační systém Raspbian “wheezy“. Raspberry Pi je novinka posledního zhruba roku a je poměrně široce používána pro studijní účely.

V první části jsou popsány techniky a principy přenosu hlasového signálu IP sítí, jeho převod do digitální podoby a následná komprese. Dále jsou popsány principy činnosti IP sítě, protokolů zajišťujících přenos dat a signalizací.

Ve druhé části jsou uvedeny možnosti volby mikroplatformy a zařízení zpracovávajících zvuk (zvuková karta, mikrofon). Takových zařízení je velmi mnoho od nejlacinějších až po velmi drahé profesionální sestavy. S ohledem na co nejvyšší kvalitu a cenovou dostupnost je uvedeno zařízení v cenové relaci úměrné ceně samotného Raspberry Pi.

Poslední část je věnována realizaci zařízení a popisu problémů, které se během tvorby vyskytly. Problém s bufferováním streamovacího klienta, kdy docházelo k přehrávání dvojnásobnou rychlostí, byl vyřešen přechodem na jinou linuxovou distribuci a volbou jiného streamovacího klienta. Komplikace s možnostmi přístupu zvukové karty k softwarové ústředně Asterisk nakonec vedly k řešení, které využívá poslech živého zvukového streamu ze stream serveru místo vyzvánění. Výhodou tohoto řešení je, že se vlastně fyzicky neuskuteční volání.

Zařízení je postaveno tak, aby umožnilo současný poslech přes protokoly SIP a HTTP, autentizaci uživatele a poslech ve dvou různých kvalitách zvukového signálu. Zatížení CPU streamovacím klientem je bohužel příliš vysoké a nedovolí pracovat s více jak jedním stereo a jedním mono mp3 streamem. Při testování na domácí síti je zpoždění zvuku zhruba 15 vteřin pro poslech přes webové stránky a zhruba minutu a 15 vteřin pro poslech přes sw voipového klienta.

## 6. Seznam literatury

- [1] Vozňák, M.: Voice over IP, Ostrava: VŠB - Technická univerzita Ostrava, 2008, ISBN 978-80-248-1828-3
- [2] Wallace, K.: VoIP bez předchozích znalostí, Brno: Computer Press, 2007, ISBN 978-80-251-1458-2
- [3] Walter, B., Janeček, V.: Telefonujeme přes Internet: Sada programů a názorný průvodce, Brno: Computer Press, 2007, ISBN 978-80-251-1631-9
- [4] Stafford, M.: Signaling and Switching for Packet Telephony, Boston: Artech House, 2004, ISBN 15-805-3736-7.
- [5] Bran, C., Jennings, C., Valin, JM.: WebRTC Codec and Media Processing Requirements, dostupné z [http://datatracker.ietf.org/doc/draft-cbran-rtcweb-codec/?include\\_text=1](http://datatracker.ietf.org/doc/draft-cbran-rtcweb-codec/?include_text=1)
- [6] Westerlund, M., Burman, B.: Codec Control for WebRTC, dostupné z [http://datatracker.ietf.org/doc/draft-westerlund-rtcweb-codec-control/?include\\_text=1](http://datatracker.ietf.org/doc/draft-westerlund-rtcweb-codec-control/?include_text=1)
- [7] Perkins, C., Westerlund, M., Ott, J.: Web Real-Time Communication (WebRTC): Media Transport and Use of RTP, dostupné z [http://datatracker.ietf.org/doc/draft-ietf-rtcweb-rtp-usage/?include\\_text=1](http://datatracker.ietf.org/doc/draft-ietf-rtcweb-rtp-usage/?include_text=1)
- [8] Jesup, R., Loreto, S., Tuexen, M.: WebRTC Data Channel Protocol, dostupné z [http://datatracker.ietf.org/doc/draft-jesup-rtcweb-data-protocol/?include\\_text=1](http://datatracker.ietf.org/doc/draft-jesup-rtcweb-data-protocol/?include_text=1)
- [9] Uhlíř, J.: Technologie hlasových komunikací, Praha: Nakladatelství ČVUT, 2007, ISBN 978-80-01-03888-8
- [10] Macich, J.: Soumrak komunikačních programů? Přichází WebRTC a SocialAPI, 2012, dostupné z <http://www.lupa.cz/clanky/soumrak-komunikacnich-programu-prichazi-webrtc-a-socialapi/>
- [11] Kovář, P.: To nejlepší pro vaše VoIP: Srovnání technologií SIP a H.323, 2008, dostupné z [http://www.intelek.cz/art\\_doc-105E7A1C373BEE63C12573FC002C37C9.html](http://www.intelek.cz/art_doc-105E7A1C373BEE63C12573FC002C37C9.html)
- [12] Wikipedia: The Free Encyclopedia, dostupné z <http://en.wikipedia.org/wiki/>
- [13] Peterka, J.: Báječný svět počítačových sítí, 2011, dostupné z [http://www.earchiv.cz/i\\_bajecnysvet.php3](http://www.earchiv.cz/i_bajecnysvet.php3)

- [14] Monroe, T.: Broadcast News, dostupné z <http://www.mactech.com/articles/mactech/Vol.18/18.03/Mar02QTToolkit/index.html>
- [15] WebRTC, dostupné z <http://www.webrtc.org>
- [16] ASM, dostupné z <http://www.asm.cz/zbozi/tc-261mxk-micropc-1x-sd-1ghz-1x-1an-3x-usb-fanless-512mb-35w-odber-vyska-20mm.html>
- [17] Odroid, dostupné z [http://www.hardkernel.com/renewal\\_2011/main.php](http://www.hardkernel.com/renewal_2011/main.php)
- [18] Cubieboard, dostupné z <http://cubieboard.org/>
- [19] Kontron, dostupné z <http://emea.kontron.com/products/boards+and+mezzanines/embedded+sbc/pitx+25+sbc/ktt20pitx.html>
- [20] ProductWiki, dostupné z <http://music.productwiki.com/behringer-uca202-u-control/>
- [21] Philips SBCME570, dostupné z [http://download.p4c.philips.com/files/s/sbcme570\\_00/sbcme570\\_00\\_pss\\_aen.pdf](http://download.p4c.philips.com/files/s/sbcme570_00/sbcme570_00_pss_aen.pdf)
- [22] Elektroakustika a ozvučení, dostupné z <http://www.elektroakustika.cz/index.html>
- [23] Raspberry Pi, dostupné z <http://www.raspberrypi.org/>
- [24] Raspi.cz, dostupné z <http://www.raspi.cz/>
- [25] Informace nejen ze světa linuxu, dostupné z <http://www.root.cz/>
- [26] Embedded Linux Wiki, dostupné z <http://elinux.org/>
- [27] Archlinux: A simple, lightweight distribution, dostupné z <https://www.archlinux.org/>
- [28] Abclinuxu, dostupné z <http://www.abclinuxu.cz>
- [29] Telegro, dostupné z <http://www.telegro.cz/>
- [31] Asterisk, dostupné z <http://www.asterisk.org/>
- [32] Voip-Info.org: A reference guide to all things VOIP, dostupné z <http://www.voip-info.org/>
- [33] DarkIce, dostupné z <http://darkice.org/>
- [34] IceCast, dostupné z <http://www.icecast.org/>
- [35] Malý, M.: HTML5 Audio ve vašich stránkách, 2010, dostupné z <http://www.zdrojak.cz/clanky/html5-audio-radio-ve-vasich-strankach/>
- [36] Janovský, D., Jak psát web: o tvorbě internetových stránek, dostupné z <http://www.jakpsatweb.cz/>

- [37] Rosebrock, E., Filson, E.: Linux, Apache, MySQL a PHP: Instalace a konfigurace prostředí pro pokročilé webové aplikace, Praha: Grada Publishing,a.s., 2005, ISBN 80-247-1260-1
- [38] Nemeth, E., Snyder, G., Hein, T.R.: Linux: Kompletní příručka administrátora, Brno: Computer Press,a.s., 2008, ISBN 978-80-251-2410-9

## **7. Elektronické Přílohy**

- Příloha 1: index.html
- Příloha 2: pozadi.jpg
- Příloha 3: znakCVUT.jpg