Fabrizio Mondo

CREARE E GESTIRE UNA WEBRAD **PROFESSIONALE**



WEBBO

Scheda sul sito >



CAMBIARE CAPPELLO SIGNIFICA CAMBIARE IDEE, AVERE UN'ALTRA VISIONE DEL MONDO.

C.G. Jung

A mia moglie Lavinia, a nostro figlio in arrivo e a quel freddo mattino di febbraio in cui per la prima volta ho conosciuto il mondo della radio su Internet

Dario Flaccovio Editore

FABRIZIO MONDO CREARE E GESTIRE UNA WEB RADIO PROFESSIONALE

ISBN 978-88-579-0275-3

© 2014 by Dario Flaccovio Editore s.r.l. - tel. 0916700686 www.darioflaccovio.it info@darioflaccovio.it

Prima edizione: aprile 2014

Mondo, Fabrizio <1985->

Creare e gestire una web radio professionale / Fabrizio Mondo. -Palermo : D. Flaccovio, 2014. ISBN 978-88-579-0275-3 1. Radio – Gestione [e] Organizzazione - Impiego [di] World Wide Web. 384.54 CDD-22 SBN Pal0267712

CIP - Biblioteca centrale della Regione siciliana "Alberto Bombace"

Stampa: Tipografia Priulla, Palermo, aprile 2014

Nomi e marchi citati sono generalmente depositati o registrati dalle rispettive case produttrici. L'editore dichiara la propria disponibilità ad adempiere agli obblighi di legge nei confronti degli aventi diritto sulle opere riprodotte.

La fotocopiatura dei libri è un reato.

Le fotocopie per uso personale del lettore possono essere effettuate nei limiti del 15% di ciascun volume/fascicolo di periodico dietro pagamento alla SIAE del compenso previsto dall'art. 68, commi 4 e 5, della legge 22 aprile 1941 n. 633. Le riproduzioni effettuate per finalità di carattere professionale, economico o commerciale o comunque per uso diverso da quello personale possono essere effettuate solo a seguito di specifica autorizzazione rilasciata dagli aventi diritto/dall'editore.

Indice

Prem	essa	Pag.	19
Capit	tolo 1 – Teoria della Web Radio	«	21
1.1. I	ntroduzione	«	21
1.2. T	eoria dei grafi	«	22
1.3. V	Veb Radio a fonte unica	«	23
1.4. V	Veb radio a fonte multipla	«	24
Capit	tolo 2 – Codifica Audio	«	27
2.1.	Come funziona MP3	«	29
	2.1.1. Anatomia di un MP3	«	31
2.2.	Come funziona WMA	«	34
2.3.	Come funziona Vorbis	«	35
	2.3.1. Dentro l'algoritmo	«	36
	2.3.2. Il bitrate peeling	«	36
	2.3.3. I container di Vorbis	«	37
2.4.	Come funziona AAC	«	37
	2.4.1. Streaming	«	38

2.5.	Come funziona FLAC	Pag.	38
	2.5.1. Passaggi di funzionamento	«	38
Capit	colo 3 - Shoutcast	«	41
3.1.	Versione 1.9	«	41
	3.1.1. Il file di configurazione	«	43
	3.1.2. L'interfaccia web di Shoutcast 1.9.7	«	55
3.2.	Versione 2.0	«	57
	3.2.1. SC_SERV_BASIC.CONF	«	60
	3.2.2. SC_SERV_DEBUG	«	61
	3.2.3. SC_SERV_PUBLIC	«	62
	3.2.4. SC_SERV_RELAY	«	63
	3.2.5. Transcoder	«	64
	3.2.6. Digital Signal Processor	«	64
	3.2.7. Input configuration Tab	«	72
	3.2.8. Soundcard mixer control Tab	«	72
3.3.	Modalità operative DSP con Winamp	«	73
	3.3.1. Connessione diretta di Winamp a Shoutcast Server v1	«	73
	3.3.2. Connessione diretta di Winamp a Shoutcast Server v2	«	74
	3.3.3. Connessione sorgente DJ tramite transcoder in un		
	Shoutcast Server 1	«	74
	3.3.4. Connessione sorgente DJ tramite transcoder in un		
	Shoutcast Server 2	«	75
3.4.	Configuration builder	«	75
3.5.	Shoutcast su Linux	«	76
	3.5.1. Sc_serv.conf	«	80
	3.5.2. Sc_trans.conf	«	89
	3.5.3. Example.lst	«	92
Capit	olo 4 - Icecast	«	95
4.1.	Impostazioni principali	«	95
	4.1.1. Icecast.xml	«	96
	4.1.2. File di configurazione	«	97
	4.1.3. Limiti	«	98
	4.1.4. Autenticazione	«	99
	4.1.5. Impostazioni delle pagine gialle	«	99
	4.1.6. Altre impostazioni	«	100
	4.1.7. Socket in ascolto	«	100
	4.1.8. Ripetitori	«	102
	4.1.9. Mount Specific Settings	«	105

4.2.	Impostazioni di percorso	Pag.	110
4.3.	Diario di bordo	«	111
4.4.	Impostazioni di sicurezza	«	113
4.5.	Interfaccia di amministrazione di Icecast 2	«	113
	4.5.1. Funzioni di amministrazione specifiche per mountpoint	«	113
	4.5.2. Funzioni generali di amministrazione	«	115
	4.5.3. Interfaccia web di amministrazione	«	116
4.6.	Statistiche del server Icecast 2	«	116
	4.6.1. Statistiche generiche	«	117
	4.6.2. Statistiche specifiche per una sorgente	«	117
4.7.	Relaying di Icecast 2	«	118
	4.7.1. Tipologie di relays	«	118
	4.7.2. Creare un relay master-slave	«	118
	4.7.3. Creare un relay single-broadcast	«	119
4.8.	Directory delle pagine gialle di Icecast 2	«	120
	4.8.1.Configurare Icecast2 per il supporto pagine gialle	«	120
	4.8.2. Configurazione per il supporto pagine gialle	«	121
4.9.	Autenticazione dell'ascoltatore Icecast 2	«	121
	4.9.1. HTPASSWD Listener Authentication	«	121
	4.9.2. Configurazione di user e password	«	122
4.10.	Conclusioni	«	124
	4.10.1. URL	«	124
4.11.	Documentazione specifica per Win32	«	127
	4.11.1. Server status tab	«	127
	4.11.2. Aggiungere statistiche alla finestra	«	128
	4.11.3. Rimuovere statistiche dal server status	«	128
	4.11.4. Modificare il file di configurazione	«	129
	4.11.5. Stats tab	«	129
Capit	olo 5 – SIAE	«	131
5.1. A	rt. 1 – Definizioni	«	131
5.2. A	rt. 2 – Oggetto della licenza	«	136
5.3. Ai	rt. 3 – Diritti Concessi	«	136
5.4. A	rt. 4 – Repertorio della SIAE	«	137
5.5. Ai	rt. 5 – Territori	«	138
5.6. A	rt. 6 – Diritti esclusi	«	139
5.7. A	rt. 7 – Riserva degli aventi diritto	«	139
5.8. A	rt. 8 – Limiti della licenza	«	140
5.9. Ai	rt. 9 – Utilizzazioni pubblicitarie	«	142

5.10.4	Art. 10 – Inizio, cessazione, variazione di attività	Pag.	143
5.11.	Art. 11 – Link tra il sito del Licenziatario e i siti di terzi	«	144
5.12.	Art. 12 – Utilizzazione di altri repertori tutelati dalla SIAE	«	145
5.13.	Art. 13 – Tipologie di Web Radio e calcolo dei compensi	«	146
	5.13.1. Web radio commerciali	«	146
	5.13.2. Web radio istituzionali e/o comunitarie	«	148
	5.13.3. Web radio personali	«	148
5.14.	Art. 14 – Estensione della Licenza all'offerta degli stessi		
	contenuti su Social Network (sperimentale)	«	149
5.15.7	Art. 15 – Termini e modalità di pagamento dei compensi	«	150
5.16.4	Art. 16 – Penali	«	151
5.17.1	Art. 17 – Report	«	152
5.18.7	Art. 18 – Controlli della SIAE	«	152
5.19.1	Art. 19 – Digital Delivery	«	154
5.20. /	Art. 20 – Termini di validità	«	155
5.21.	Art. 21 – Risoluzione del contratto di Licenza	«	156
5.22.	Art. 22 – Legge applicabile e Foro competente	«	157
5.23. /	Art. 23 – Dichiarazione privacy	«	157
Capit	olo 6 – SAM Broadcaster	«	159
6.1.	Menu File	«	159
6.2.	Menu Player	«	160
6.3.	Menu General	«	161
6.4.	Menu <i>Window</i>	«	161
	6.4.1. Deck	«	161
6.5.	Player display	«	162
6.6.	Controlli ausiliari	«	163
6.7.	Playlist	«	164
	6.7.1. Comprendere le categorie di playlist	«	164
6.8.	Queue	«	168
	6.8.1. Queue Display	«	168
	6.8.2. Aggiungere e rimuovere dalla coda	«	169
	6.8.3. Scorciatoie da tastiera	«	170
6.9.	History	«	170
	6.9.1. Scorciatoie da tastiera	«	171
6.10.	SAM requests	«	171
	6.10.1. Request Display	«	171
	6.10.2. Scorciatoie da tastiera	«	171
6.11.	Voice FX	«	172

	6.11.1. Voice FX Display	Pag.	172
	6.11.2. Configurazione	«	173
	6.11.3. Utilizzo di Voice FX	«	174
	6.11.4. Scorciatoie da tastiera	«	175
6.12.	Volume	«	175
6.13.	Sound FX	«	175
	6.13.1. Sound FX display	«	176
	6.13.2. Sound FX control	«	176
6.14.	Fade Control	«	177
	6.14.1. Crossfading manuale	«	177
	6.14.2. Fading manuale	«	178
	6.14.3. Tempo adjust (pitch control)	«	178
	6.14.4. Smooth Beat Fade	«	179
	6.14.5. Beat fade	«	179
6.15.	SAM Voice-tracking	«	180
	6.15.1. Configurazione del voice-tracking	«	180
	6.15.2. Voice-tracking: comandi principali	«	181
6.16.	StreamAds Control	«	181
	6.16.1. Control buttons	«	182
6.17.	Encoders	«	183
	6.17.1. Encoders Display	«	183
	6.17.2. Interfaccia di scripting	«	184
	6.17.3. Come aggiungere un encoder	«	185
6.18.	Statistic Relays	«	189
	6.18.1. Statistic Relays Control	«	189
	6.18.2. Come aggiungere un relay di statistiche	«	190
	6.18.3. Statistic relays graph	«	193
6.19.	FTP log	«	194
6.20.	PAL script	«	195
6.21.	PCI Max Ultra	«	196
6.22.	Event Scheduler	«	196
	6.22.1. Event Action Tab	«	197
	6.22.2. Schedule Times Tab	«	197
	6.22.3. Event log	«	198
	6.22.4. Event log display	«	198
6.23.	Clock	«	199
6.24.	Menu Layout	«	199
6.25.	Menu Tools	«	199
6.26.	Menu Help	«	200

6.27.	Toolbars	Pag.	201
6.28.	Configurazione di SAM Broadcaster	«	202
6.29.	Member Details	«	202
6.30.	Station Details	«	202
6.31.	Audio Mixer Pipeline	«	202
6.32.	Crossfading	«	203
	6.32.1. Configurazione di base	«	203
	6.32.2. Configurazione del crossfading	«	205
6.33.	Gap killer	«	206
6.34.	Playlist Rotation Rules	«	207
	6.34.1. Playlist Rotation Logic Modules	«	209
6.35.	Request Policy	«	210
	6.35.1. Regole di richiesta	«	211
	6.35.2. Vantaggi e svantaggi del ritardo di richiesta	«	212
6.36.	General	«	214
	6.36.1. Configurazione basilare	«	214
6.37.	HTML Output	«	216
	6.37.1. General HTML output	«	217
	6.37.2. Playlist HTML output	«	218
	6.37.3. Upload file via FTP	«	218
6.38.	StreamAds Providers	«	218
6.39.	SAM StreamAds Logic	«	220
6.40.	Registration Key	«	221
6.41.	Equalizzatore	«	221
6.42.	AGC (Automatic Gain Control)	«	222
6.43.	DSP (Digital Sound Processing)	«	224
6.44.	Output	«	225
	6.44.1. Air Output Settings	«	225
	6.44.2. Cue output settings	«	226
	6.44.3. Buffer settings	«	226
6.45.	SAM Shortcut Keys	«	227
	6.45.1. Scorciatoie globali	«	227
	6.45.2. Scorciatoie di finestra	«	227
6.46.	Editor delle informazioni dei brani	«	230
6.47.	Tab details	«	230
6.48.	Tab Settings	«	230
	6.48.1. Cue point	«	230
	6.48.2. Come settare un cue point	«	231
	6.48.3. Beat-per-minute	«	231

	6.48.4. Altre opzioni	Pag.	232
	6.48.5. Opzioni di applicazione	«	232
	6.48.6. Tab Fading	«	232
6.49.	Troubleshooting	«	232
	6.49.1. Touch Aborted	«	232
	6.49.2. Audio Fast Forwarding	«	232
	6.49.3. SAM was unable to queue a song after 20 tries	«	233
	6.49.4. Unable to send data fast enough	«	234
	6.49.5. Codici numerici d'errore	«	234
6.50.	Sezione vocale	«	238
6.51.	Template per il web	«	238
	6.51.1playing.html	«	239
	6.51.2songinfo.html	«	242
6.52.	Approfondimento sui PAL Scripts	«	246
	6.52.1. Introduzione	«	246
	6.52.2. Avvertenze per l'utilizzo degli script PAL	«	246
	6.52.3. Come creare un PAL script	«	246
	6.52.4. Inserire un commento in uno script	«	247
	6.52.5. Comandi utili per il debugging	«	247
	6.52.6. Variabili e costanti	«	248
	6.52.7. Matematica del PAL	«	249
	6.52.8. Gestione delle stringhe	«	251
	6.52.9. Altre funzioni di interesse per le stringhe	«	252
	6.52.10. Gestione del tempo	«	253
	6.52.11. La logica nel PAL	«	256
	6.52.12. Costrutti principali	«	257
	6.52.13. Comandi di attesa	«	259
	6.52.14. Velocizzare l'esecuzione degli script	«	262
	6.52.15. Gli oggetti	«	262
	6.52.16. I principali oggetti	«	264
	6.52.17. Funzioni di utilità	«	268
	6.52.18. LinerAdder	«	269
	6.52.19. Funzioni di utilità - Curtime	«	273
	6.52.20. Funzioni di utilità – Gestione del volume	«	274
	6.52.21. Encoder e relay di statistiche	«	275
	6.52.22. Tecniche di scripting avanzate	«	280
	6.52.23. Scaricare file	«	283
	6.52.24. Funzioni per maneggiare i nomi dei file	«	283
	6.52.25. Funzioni per il file system	«	284

	6.52.26. Maneggiare indirizzi remoti	Pag.	287
	6.52.27. Maneggiare le categorie	«	288
	6.52.28. Selezionare brani	«	290
Capit	tolo 7 - Zararadio	«	291
7.1.	Opzioni settabili	«	293
	7.1.1. Fade	«	293
	7.1.2. Satellite	«	294
	7.1.3. Uscite	«	294
	7.1.4. Registro	«	295
	7.1.5. Password	«	296
	7.1.6. Rilevatore di silenzio	«	297
	7.1.7. Mixer	«	298
	7.1.8. TTH	«	298
	7.1.9. Explorer	«	299
	7.1.10. Tags	«	299
	7.1.11. Talk Over	«	300
	7.1.12. Generale	«	301
Canif	tolo 8 - Windows Media Encoder	u	303
8 1	Onzioni	"	307
0.1.	8 1 1 Sources	"	308
	8.1.2. Output	«	309
	8.1.3. Compression	«	310
	8.1.4. Attributes	«	313
	8.1.5. Plug-ins	«	314
	8.1.6. Security	«	315
	8.1.7. Advanced	«	316
8.2.	Informazioni di trasmissione	«	317
	8.2.1. General	«	317
	8.2.2. Statistics	«	319
	8.2.3. Server	«	320
	8.2.4. Connections	«	321
	8.2.5. Event Log e EDL	«	321
Canit	tolo 9 - IDIC	"	325
9.1	Trasmettere con IDIC	"	327
<i>/</i> 1 1 1	9.1.1. Sezione Connection	«	328
	9.1.2. Sezione Encoding	«	329

	9.1.3. Sezione Stream Info	Pag.	329
	9.1.4. Sezione Record	«	330
9.2.	Preferenze	«	330
	9.2.1. General	«	330
	9.2.2. Microphone	«	332
	9.2.3. X-Chat	«	332
	9.2.4. Jack Ports	«	333
	9.2.5. Event	«	333
9.3.	TroubleShooting	«	334
Capit	olo 10 - Metafile	«	337
10.1.	PLS	«	337
10.2.	RAM	«	338
10.3.	ASX	«	338
Capit	olo 11 - SCF	«	339
11.1.	Il contratto per web radio amatoriale	«	340
	11.1.1. Art. 1 – Definizioni	«	341
	11.1.2. Art. 2 – Oggetto	«	343
	11.1.3. Art. 3 – Modalità di utilizzazione delle registrazioni		
	fonografiche	«	344
	11.1.4. Art. 4 – Compenso	«	348
	11.1.5. Art. 5 – Termini e modalità di pagamento	«	348
	11.1.6. Art. 6 – Controlli	«	349
	11.1.7. Art. 7 – Ulteriori obblighi del Licenziatario	«	349
	11.1.8. Art. 8 – Dichiarazioni e manleve	«	351
	11.1.9. Art. 9 – Durata	«	351
	11.1.10. Art. 10 – Risoluzione in caso di inadempimento	«	351
	11.1.11. Art. 11 – Legge applicabile e foro competente	«	352
	11.1.12. Art. 12 – Disposizioni generali	«	352
	11.1.13. Art. 13 – Trattamento dei dati personali	«	353
	11.1.14. Allegato A	«	353
11.2.	Il contratto per web radio istituzionale	«	355
11.3.	Il contratto per web radio commerciale	«	355
	11.3.1. Compenso Streaming	«	356
	11.3.2. Minimo Garantito	«	358
	11.3.3. Rendicontazione	«	358
	11.3.4. Termini e modalità di pagamento	«	360

Capitolo 12 - Streamripper		Pag.	361
12.1.	StreamRipper come plug-in per Winamp	«	362
12.2.	Configurazione del plug-in	«	363
	12.2.1. Connection	«	363
	12.2.2. File	«	363
	12.2.3. Pattern	«	364
12.3.	StreamRipper come standalone su GNU/Linux	«	365
12.4.	Manuale tecnico di Streamripper	«	365
	12.4.1. Individuazione dello Splitpoint	«	366
	12.4.2. Splitting standard	«	366
	12.4.3. Separazione basata sul silenzio	«	366
Capit	olo 13 - GnuMP3d	«	369
13.1.	Installazione su Windows	«	369
13.2.	Installazione su Linux	«	370
	13.2.1. Compilazione	«	370
	13.2.2. Tramite APT	«	370
	13.2.3. Dipendenze	«	370
13.3.	Configurazione di GnuMP3d su Linux	«	371
	13.3.1. File.types	«	371
	13.3.2. Mime.types	«	373
	13.3.3. Gnump3d.conf	«	374
Capit	olo 14 - Spreaker	«	405
14.1.	Interfaccia della consolle	«	407
Capit	olo 15 - DirEttore	«	411
15.1.	Requisiti minimi	«	411
15.2.	Installazione della regia	«	411
15.3.	File audio supportati	«	413
15.4.	Caratteristiche principali e definizioni	«	413
15.5.	La finestra principale	«	414
	15.5.1. Il pannello dei player	«	415
	15.5.2. Il pannello dei contatori	«	416
	15.5.3. Il volumetro principale	«	420
	15.5.4. Primo gruppo comandi	«	421
	15.5.5. Secondo gruppo comandi	«	426
	15.5.6. Terzo gruppo di comandi	«	428
	15.5.7. Playlist e rotazione	«	428

15.6.	Il preascolto	Pag.	430
15.7.	Editor delle informazioni dei brani	«	431
15.8.	Configurazione dei player degli effetti istantanei	«	432
15.9.	Configurazione della rotazione	«	433
15.10.	Configurazione della playlist	«	435
15.11.	Schedulatore degli eventi	«	438
15.12.	Pianificazione delle pubblicità	«	442
15.13.	Editor delle sequenze	«	444
15.14.	Editor dei cue point	«	445
15.15.	Configurazione	«	447
	15.15.1. Parametri generali	«	447
	15.15.2. Track defaults tab	«	448
	15.15.3. Track Info tab	«	449
	15.15.4. Logging tab	«	450
	15.15.5. Rotation tab	«	451
	15.15.6. Panic tab	«	452
	15.15.7. Sweeper and voice-tracks tab	«	453
	15.15.8. Time announcement tab	«	455
	15.15.9. View tab	«	455
	15.15.10. Audio cards tab	«	456
	15.15.11. Compressor tab	«	458
	15.15.12. Equalizer tab	«	458
	15.15.13. Password tab	«	459
	15.15.14. Remote control tab	«	461
	15.15.15. Automation behaviour tab	«	462
	15.15.16. VST effect tab	«	463
15.16.	Come avviare l'automazione	«	464
15.17.	Scorciatoie da tastiera	«	464
15.18.	I file generati dalla regia	«	465
	15.18.1. Listato di un file playlist	«	465
	15.18.2. Listato di un file rotazione	«	466
	15.18.3. Listato di una collezione di eventi	«	467
	15.18.4. Listato del piano pubblicitario	«	467
	15.18.5. Listato delle sequenze	«	468
	15.18.6. Modalità di salvataggio dei cue point	«	469
	15.18.7. Listato dei player degli effetti istantanei	«	470
Capito	010 16 - GISS.tv	«	471
16.1.	Configurazione del canale	«	472

Capitolo 17 - Caster.fm		Pag.	473
17.1.	Tab server	«	474
17.2.	Tab website	«	474
17.3.	Tab tutorials	«	476
17.4.	Tab downloads ed embed	«	477
Capito	olo 18 - Edcast	«	479
18.1.	Menu di configurazione	«	481
	18.1.1. Basic Settings	«	481
	18.1.2. Advanced settings	«	483
18.2.	Metadata	«	483
Capito	olo 19 - MB Studio	«	485
19.1.	Audio	«	486
	19.1.1. Scheda Audio	«	486
	19.1.2. Opzioni	«	489
	19.1.3. Mixaggi	«	490
19.2.	Opzioni	«	490
	19.2.1. Opzioni	«	490
	19.2.2. Altro	«	492
	19.2.3. Annuncio	«	494
	19.2.4. Caratteri	«	495
	19.2.5. RSS Feed	«	495
19.3.	Canzoni	«	496
19.4.	Cartelle	«	498
19.5.	Voci sintetiche	«	499
19.6.	Modem	«	500
19.7.	Rete	«	502
	19.7.1. Cartella di lavoro MB Studio	«	502
	19.7.2. Connessioni TCPIP	«	503
	19.7.3. Split	«	504
	19.7.4. Titoli esterni	«	505
	19.7.5. MB Live	«	506
	19.7.6. Altro	«	507
19.8.	RDS Pira	«	508
19.9.	Webcast	«	508
	19.9.1. Tab Cattura	«	508
	19.9.2. Tab DSP Plugin	«	509
	19.9.3. Tab Testo metadata	«	509

	19.9.4. Tab Encoder (da 1 a 4)	Pag.	510
	19.9.5. Tab 90 giorni	«	511
	19.9.6. Tab registratore	«	511
19.10	. Email	«	514
	19.10.1. POP3	«	514
	19.10.2. SMTP	«	514
19.11	Pubblicazioni	«	515
	19.11.1. Playlist	«	515
	19.11.2. Messaggi	«	516
	19.11.3. Cartella immagini	«	517
	19.11.4. FTP Upload	«	517
Capit	olo 20 - Hardware	«	519
20.1.	Mixer	«	519
	20.1.1. Canali di ingresso	«	520
	20.1.2. Connettori di ingresso	«	521
	20.1.3. Stadio di ingresso	«	521
	20.1.4. Equalizzatore	«	521
	20.1.5. Mandate monitor	«	521
	20.1.6. Mandate effetti	«	522
	20.1.7. Selettore dei gruppi	«	522
	20.1.8. Pan-pot	«	522
	20.1.9. Potenziometro (Fader)	«	522
	20.1.10. Canali di uscita	«	522
	20.1.11. Master	«	522
	20.1.12. AUX	«	523
	20.1.13. Effetti	«	523
	20.1.14. Gruppi	«	523
20.2.	Microfoni	«	523
20.3.	Amplificatore per cuffia	«	524
20.4.	Cavi e connettori	«	525
	20.4.1. Connettore XLR	«	525
	20.4.2. Connettore Jack	«	526
	20.4.3. Connettore RCA	«	527
Glossa	rio	«	531
Iscrizi	one alla mailing list	«	551

Premessa

Creare e gestire una web radio professionale è un testo sviluppato per fornire in un unico volume tutte le indicazioni necessarie a qualsiasi persona, associazione o società che voglia realizzare una stazione radiofonica per trasmettere la propria voce ed i propri contenuti. Questo libro svolge contemporaneamente il ruolo di testo formativo e guida da consultare all'occasione. Il volume racchiude tutto quello che può servire al regista, allo speaker o anche al semplice appassionato non solo per creare una radio su Internet partendo da zero, ma anche per curarla, gestirla e migliorarla costantemente.

Ciò che ha spinto l'Autore alla stesura di questo testo unico del regista radiofonico è la volontà di raccogliere in un singolo volume tutto lo stato dell'arte della radiofonia web italiana. Con un occhio al presente ed un occhio al futuro, il testo permette di non confondersi nel cercare le informazioni di cui si necessita durante la fase di nascita o crescita della propria web radio consigliando degli standard di lavoro e guidando il lettore in un campo che richiede notevole tempismo e preparazione. Il volume tratta tre macroargomenti: teoria della web radio, software/ hardware e normative legali. La teoria consente di apprendere come strutturare un progetto di stazione radiofonica e suggerisce al lettore le corrette scelte che porteranno alla creazione della radio. Attraverso lo studio dei programmi applicativi e delle attrezzature discusse si potrà scegliere di quali strumenti fornire la web radio per dotarla di standard professionali. Tramite il commento alle normative legali si potrà trasmettere la propria musica nel pieno rispetto delle regole vigenti. È altresì presente un glossario finalizzato a chiarire eventuali dubbi sui termini tecnici frequentemente usati nella trattazione.

Capitolo 1 - Teoria della Web Radio

1.1. Introduzione

Non è possibile datare ufficialmente la nascita delle web radio nel mondo, né è possibile farlo in un singolo paese quale l'Italia. *Web Radio* identifica più una tipologia di comunicazione che una particolare tecnologia: da qui l'impossibilità di identificare una data certa.

Si può provare a considerare come anno zero il 1995, anno in cui Rob Glaser realizza la prima *release* del software RealAudio, seguito a ruota da Microsoft Media Services. Si arrivò quindi a Shoutcast, Icecast e decine di altri server di streaming. Parallelamente, si sviluppavano regie radiofoniche più o meno complesse. Non si può ovviamente datare la voglia nata nella popolazione di trasmettere tramite Internet la propria musica e la propria voce. L'impossibilità tecnica e burocratica di creare una stazione radiofonica tradizionale a basso costo ha certamente aiutato questa tecnologia a svilupparsi.

La facilità di porsi come locali e globali contemporaneamente ha dato la spinta finale a radio e TV online, che conquistano sempre più ascoltatori togliendoli ai media tradizionali.

La principale critica mossa anni fa alle previsioni di sviluppo del fenomeno web radio nasceva dal riconoscimento dell'impossibilità di ascolto in movimento.

Le radio tradizionali basavano i propri ascolti anche sugli ascoltatori occasionali che viaggiavano in auto o che venivano fidelizzati da un singolo programma di una singola emittente.

Il problema ascolto in movimento è stato rimosso dalla tecnologia che consente la connessione ad Internet anche muovendosi ad alta velocità. Le web autoradio, che hanno dato il definitivo colpo di grazia alle radio in modulazione di frequenza, non sono più novità.

La qualità dei programmi e la loro adeguata pubblicizzazione diventeranno quindi gli unici criteri su cui basare l'*audience* di un'emittente, in tutto il mondo.

1.2. Teoria dei grafi

La teoria delle web radio è profondamente legata alla teoria dei grafi e alla teoria delle reti di calcolatori. Una web radio (o più precisamente un terminale e la sua banda in upload) è assimilabile alla radice di un grafo aciclico connesso, ovvero di un albero N-ario di altezza 1. Ovvero si ha una situazione simile a quella rappresentata nella figura 1.1.



Figura 1.1. Schema di architettura centralizzata client/server

Il nodo S è il computer che trasmette, detto *server*, mentre i nodi C sono i computer o gli apparecchi che ricevono il flusso, chiamati *client*. Il server è la "radice" dell'albero, mentre i client sono le "foglie". L'altezza dell'albero è 1, come precedentemente detto. Il numero massimo di "foglie" della radice, ovvero di client che possono ascoltare la radio è dato (teoricamente) dalla seguente formula:

Client Totali =
$$\frac{\text{Upload}}{\text{Bitrate}}$$

Ad esempio, se si ha una banda in upload di 1 MegaBit per secondo, e si imposta un bitrate di 64 kbps, si potrà teoricamente essere ascoltati al massimo da 16 client. Uno dei modi con cui è possibile aumentare il numero massimo di ascoltatori è quello di aumentare la banda in upload, ma ne esistono degli altri. Ad esempio, diminuire il bitrate: tuttavia, ciò andrebbe a discapito della qualità d'ascolto.

Il terzo modo è quello di replicare la fonte, ed utilizzare i ripetitori, comunemente detti relay.

1.3. Web Radio a fonte unica

Una radio è a fonte unica se ogni client o ogni relay di livello inferiore al primo riceve il flusso da un solo ripetitore:

- il nodo S è il server, il computer principale al quale arriva il flusso dalla sorgente;
- i nodi R sono dei client, ma sono anche dei relay, ovvero dei ripetitori di segnale ai quali si appoggiano altri client;
- i nodi C sono client che non svolgono il ruolo di ripetitori.



Figura 1.2. Albero di una web radio a fonte unica

Si noti che ogni client è promuovibile a relay, ma vi possono essere relay a cui nessuno si è connesso.

Questi alberi hanno altezza pari al massimo numero di relay che si devono attraversare prima di arrivare alla fonte del segnale, più uno. Questa configurazione, se da un lato diminuisce la domanda di banda imposta al server, distribuendola anche ai relay, d'altra parte aumenta in funzione logaritmica il ritardo di propagazione. Il computer A trasmette al computer B che ripete a sua volta al computer C che ripete a sua volta al computer D, ovvero:

In questo caso il ritardo di propagazione vale 4 volte il ritardo di propagazione di una trasmissione diretta. Il buon senso ci ricorda, in ogni caso, che la situazione lineare di cui sopra non ha nessuna utilità pratica. Se, invece, si ipotizza che la radice e tutti i suoi relay abbiano al massimo due ascoltatori, che possono a loro volta fare da relay, allora si crea un albero binario, ed il ritardo di propagazione stavolta sarebbe sottostimato da

 $log_2 x$

Con *x* che rappresenta il numero di ascoltatori contemporaneamente connessi. La base del logaritmo dipende dal numero di ascoltatori che può servire ciascun relay. Se questo numero aumenta, aumenterà la base del logaritmo e, di conseguenza, diminuirà il numero di relay che dovrà attraversare un client per sentire il flusso audio.

Se si riesce a distribuire bene il numero di computer che fungono da relay con il numero di client che richiedono il flusso, allora il ritardo di propagazione crescerà in funzione del logaritmo del numero dei client contemporaneamente connessi alla radio.

1.4. Web radio a fonte multipla

Una web radio a fonte multipla, a differenza di una a fonte singola, prevede che i client possano ricevere il flusso da più fonti contemporaneamente.



Figura 1.3. Albero di una web radio a fonte multipla

A prima vista l'immagine potrà sembrare un grafo ciclico connesso; in realtà non è affatto ciclico, in quanto è orientato. I client non rimandano nulla ai relay, quindi non è possibile che un relay invii qualcosa ad un altro relay tramite un client. Questa situazione è la migliore in tutti i campi, ed è di fatto lo schema del *peer to peer*.

Se invece ci si limita a scaricare senza condividere si ha solo la qualifica di client, o *dead end*. Questo schema multi-connesso è una buona soluzione al problema banda, in quanto si effettua una partizione del carico nel maggior numero di computer possibili. La soluzione prevista è però parecchio complessa da implementare in pratica, se non impossibile.

Si supponga di essere un semplice client, che vuole sentire una web radio che ha implementato lo schema a fonte multipla. La radio permette l'ascolto passando attraverso 4 relay, perché sono tutti occupati ai livelli superiori. Il ritardo è quindi pari a 4.

Ad un certo punto, si libera uno "spazio" su un relay al livello 2, utilizzabile dall'ascoltatore. Adesso però arriveranno parti dello stesso flusso, sfasate tra loro di una certa quantità di tempo che è data dalla differenza dei ritardi.

Anche se fosse possibile segmentare il flusso non si potrebbe non essere vincolati dai diversi ritardi qualora si volesse ottenere fluidità d'ascolto e il minimo ritardo possibile.

Purtroppo, non è nemmeno possibile usufruire di più relay sullo stesso livello, perché i ritardi di propagazione non dipendono soltanto dal livello, ma anche da molti altri fattori.

Capitolo 2 – Codifica Audio

Uno dei processi fondamentali che si instaurano durante la trasmissione radiofonica sul web è l'*encoding*. La codifica è quel passaggio che consente di trasformare un segnale di un tipo in un segnale di un altro tipo che presenta determinate caratteristiche. Il caso più frequente è la conversione da un segnale analogico, quale può essere il suono di un disco in vinile, ad un segnale digitale, quale può essere un MP3.

Negli studi radiofonici assistiamo alla convivenza temporale di segnali aventi caratteristiche differenti, quali la musica (che solitamente è già in formato digitale, ma non è detto a priori) e la stessa voce dello speaker. L'encoder non fa altro che convertire e campionare questi segnali eterogenei in un segnale ben definito. Il segnale risultante può essere di diverse tipologie, a secondo che la campionatura venga effettuata da un tipo di algoritmo piuttosto che da un altro o se alla campionatura venga o meno associata una compressione.

Esistono quindi una infinità di codifiche audio; non tutte sono di largo utilizzo per la trasmissione su Internet di flussi audio, ma sicuramente è comodo conoscerle. I codec audio si dividono in tre macrocategorie: non compressi, compressi con perdite (lossy) e compressi senza perdite (lossless).

Tra i codec non compressi, i più famosi sono WAV e AIFF; tra i compressi senza perdite troviamo FLAC e ALE, mentre tra i compressi con perdite si ha MP3, AAC e Vorbis. Qualsiasi codifica audio, a prescindere dalla categoria, ha delle caratteristiche che consentono di calcolare la grandezza in byte del file risultante. Tale grandezza è calcolata in base alla seguente regola

$$C \times Ca \times Ris \times T$$

dove *C* sta per *numero canali* (grandezza adimensionale), *Ca* sta per *frequenza di campionamento* (si misura in Hertz), *Ris* sta per *Risoluzione del Campione* (si misura in byte) e *T* sta per *durata* (si misura in secondi). La grandezza di un file WAV, in stereo, a 44100 Hz di frequenza di campionamento a 16 bit, della durata di un minuto è quindi pari a

 $2 \times 44100 \text{ s}^{-1} \times 16 \text{ bit} \times 60 \text{ sec} = 84672000 \text{ bit}$

che corrispondono a 10584000 byte, ovvero poco più di 10 MB. Un minuto di musica codificata in formato WAV richiede quindi dieci megabyte: un valore eccessivamente alto per qualsiasi trasmissione, perché corrisponderebbe ad una trasmissione alla velocità di 1378 kbps, più di dieci volte lo standard del bitrate di una comune web radio. Ecco perché sono intervenute attività di compressione a corredo della codifica. La maggior parte delle codifiche *lossy* tende ad eliminare le frequenze dei suoni impercettibili all'orecchio umano, concentrandosi quindi sulla banda di suoni che va dai 16 ai 20000 Hertz. Oltre a ciò, MP3 ed altre codifiche applicano algoritmi che filtrano il segnale originale, eliminando suoni a loro volta coperti da suoni più forti e che il nostro orecchio non potrebbe correttamente percepire. L'utilizzo di strumenti matematici quali la trasformata di Fourier o la codifica di Huffman consentono di diminuire ulteriormente la grandezza del file finale, che risulta abbattuto anche del 90% rispetto all'originale file WAV.

La codifica *lossy* è però irreversibile, al contrario di quanto non lo sia la *lossless*. Un file MP3 non potrà mai essere ricondotto al file WAV originale e meno ancora al reale segnale analogico.

2.1. Come funziona MP3

MP3 è l'acronimo di MPEG 1 layer 3. MPEG è a sua volta l'acronimo di *Moving Picture Experts Group*, il nome ufficiale di una commissione formata dall'International Standards Organization (ISO) e dalla commissione Internazionale Elettrotecnica (IEC) per sviluppare standard internazionali per un *encoding* efficiente di video ed audio ad alta qualità.

La differenza tra MP3 e la classica codifica PCM vista precedentemente consiste nella tipologia di compressione e di rimozione dei suoni che non vengono percepiti correttamente dall'orecchio umano.

La psicoacustica è la branca dell'acustica che studia la percezione del suono da parte dell'uomo. Si è scoperto, ad esempio, che raddoppiare l'ampiezza di un suono non è automaticamente correlato alla sensazione di un suono forte il doppio. La natura umana prevede che alcune fasce di frequenze tra quelle udibili siano maggiormente considerate rispetto ad altre che sono ai margini dell'intervallo, principalmente per una questione di probabilità d'ascolto e di interesse.

Sfruttare la limitatezza del nostro apparato uditivo ha permesso la creazione non soltanto di MP3, ma in generale di tutti gli algoritmi di codifica *lossy*.

Il *perceptual coding* di MP3 si occupa esattamente di codificare digitalmente non per come esso è, come fa PCM, ma per come esso suona, permettendo di diminuire anche del 90% il peso finale del file codificato. Ci si deve chiedere quindi cosa MP3 tolga dal file PCM originario. Per rispondere a questa domanda occorre introdurre i concetti di irrilevanza e di ridondanza.

Un dato è ridondante se è possibile rimuoverlo dal file audio senza grosse perdite di qualità. L'esempio più classico è il limite di Nyquist.

Se stiamo campionando il segnale a 44100 hertz, il nostro limite di Nyquist, secondo il teorema del campionamento di Nyquist-Shannon, è di 22050 hertz, per cui tutto quel suono che è oltre questa frequenza non verrà contemplato e, anche se lo fosse, non sarebbe correttamente interpretato dall'uomo: per cui viene semplicemente rimosso.

Un dato è invece irrilevante se viene mascherato dall'orecchio umano con un altro suono. È il caso di suoni aventi stessa forma d'onda di un altro suono, ma con ampiezza raddoppiata. Essi saranno mascherati dal suono ad ampiezza più alta. Il mascheramento attribuisce ai suoni irrilevanti un numero di bit inferiore rispetto ai suoni dominanti, creando una distorsione che però risulta confinata e non percettibile all'orecchio umano.

Il primo passaggio del processo di codifica comporta prendere diverse piccole sezioni del file PCM e processarle con quello che viene chiamato "filtro analitico". Lo standard MPEG 1 non specifica esattamente come questo filtro debba essere costruito, ma solo cosa dovrebbe fare. MP3 tipicamente attua una serie di algoritmi matematici quali la *trasformata veloce di Fourier* o la *trasformata discreta del coseno* per fare questa suddivisione. Ogni sezione viene quindi divisa in 32 sottobande, che rappresentano differenti parti dello spettro di frequenza del segnale PCM originario.

Una qualsiasi sezione di un file PCM contiene una miscela di differenti suoni più o meno predominanti. Dividendo in sottobande, MP3 è in grado di ordinare differenti suoni in base al contenuto in frequenza e dare un ordine di priorità in base al modello psicoacustico.

Il passaggio successivo del processo è la ricostruzione delle sottobande all'interno di frame. L'encoder esamina il contenuto di questi frame e cerca di determinare dove avvenga un mascheramento (sia nel dominio della frequenza sia in quello del tempo) e dove possano essere distorti senza particolari problemi per il suono finale. L'encoder calcola il valore di "Mask to noise" di ciascun frame e usa questa informazione nel passaggio finale del processo, l'allocazione dei bit.

Durante l'allocazione dei bit l'encoder decide, in base alla priorità definita precedentemente, come spartire la torta. La torta è composta da un numero di bit pari al bitrate che si è imposto al file. Un bitrate di 128kbps fornirà una torta di 128000 bit per ogni secondo del brano.

L'ultimo processo è la codifica di Huffmann, che consente di mappare le informazioni in "simboli". I simboli più presenti all'interno del file saranno codificati con un numero di bit più piccolo, che va ingrandendosi in base alla minore presenza del simbolo nel brano.

Quando la codifica è terminata vengono salvati tutti i frame, con alcuni byte dovuti agli header. Una volta salvati i frame viene creato il file MP3. Il file risultante può essere letto da un player, che deve limitarsi a ricostruire un segnale analogico partendo dai bit allocati ai vari frame; è un lavoro molto meno pesante rispetto a quello fatto dall'encoder, il che giustifica anche l'esistenza di tantissimi player MP3 diversi, ma molti meno encoder.

2.1.1. Anatomia di un MP3

Un file MP3 è così strutturato:

- frame 1
 - header del frame 1
 - corpo del frame 1
- frame 2
 - header del frame 2
 - corpo del frame 2
- frame N
 - header del frame N
 - corpo del frame N
- metadati (tag ID3).

HEADER DI UN FRAME

Ogni MP3 è segmentato in milioni di frame, ciascuno contenente dati audio per una frazione di secondo. All'inizio di ogni frame c'è l'header, che contiene 32 bit di metadati relativi al frame stesso.

ld	Scopo	N. di bit
A	Blocco di sincronizzazione	11
В	Versione di MPEG	2
С	Layer di MPEG	2
D	Bit di controllo	1
E	Indice del bitrate (esiste una tabella adatta alla versione di MPEG utilizzata)	4
F	Frequenza di campionamento (sempre tramite tabella)	2
G	Bit di padding per informazioni mancanti	1
Н	Bit privato per eventuali eventi correlati	1
Ι	Canali settati (joint, stereo tradizionale)	2
J	Modalità (usato solo con joint stereo)	2
K	Copyright	1
L	Originale/Copia	1
Μ	Enfasi (opzione obsoleta)	2

Tabella 2.1. Contenuto dell'header di un frame

BLOCCO DI SINCRONIZZAZIONE

11 di questi 32 bit sono chiamati "blocco di sincronizzazione". Questo blocco consente il cosiddetto *seeking*, ovvero consente ai player di cercare e fermarsi in un determinato frame, identificandolo come tale. Un player non può però, sulla base di soli 11 bits, determinare se un file è un MP3 valido, per cui occorrerà valutarne altri nel file o vederne una sequenza.

VERSIONE E PROTEZIONE

Dopo il blocco di sincronizzazione ci sono i bit dedicati alla versione e alla protezione. Se il bit di protezione non è settato, una checksum di 16 bit verrà messa prima dell'inizio del blocco di dati audio del frame.

BITRATE E FREQUENZA

Il campo bitrate definisce il bitrate del frame, che può non essere costante per tutti i frame del brano (come nel caso della codifica VBR). Segue subito il blocco frequenza di campionamento che dipende dalla versione di MPEG in uso.

PADD1NG

Il bit di padding serve ad allineare i frame che abbiano discrepanze tra di loro. Se un frame è di X byte ed un altro di X-1 il bit di padding serve a compensare la loro diversità.

CANALI SETTATI E MODALITÀ

Il campo modalità indica se il frame è mono o stereo e permette di settare le opzioni della modalità stereo, joint stereo o mono. I bit successivi servono a fornire maggiori informazioni nel caso in cui si sia scelta la modalità *joint stereo*.

COPYRIGHT E PRIVACY

Il bit di copyright indica che è illegale copiare l'MP3 al quale il frame appartiene mentre il bit di privacy può essere usato da specifiche applicazioni per causare delle azioni.

ENFAS1

Il campo *enfasi* è usato come flag in caso che il corrispondente bit sia stato settato nel file originario. È un bit ormai in disuso.

TAG 1D3

I tag ID3 (*identify an MP3*) possono essere inseriti all'inizio oppure alla fine del file MP3 che devono identificare. Contengono al loro interno informazioni sull'artista, sul titolo, sul copyright e i termini di utilizzo, un'immagine che può essere visualizzata sul player e commenti.

Attualmente ci sono due varianti di ID3: la versione 1 e la versione 2. La differenza tra le due versioni risiede soltanto nella quantità di informazioni memorizzabili. I tag ID3 versione 2 compaiono molto più spesso all'inizio che non al la fine del file. Il motivo è presto detto: quando un MP3 viene trasmesso, il player deve sapere in anticipo il contenuto del file che sta eseguendo per fornire le informazioni all'ascoltatore.

JOINT STEREO

In modalità *joint stereo* i due canali, il destro ed il sinistro, non sono indipendenti come ci si potrebbe immaginare: in realtà sono realizzati in modo che uno sia la somma dei due canali, mentre l'altro ne sia la differenza. Un decodificatore che riceve due canali in joint stereo può comunque ricostruire i due canali senza perdite grazie a questa relazione matematica:

detto A il canale somma (X + Y) e B il canale differenza (X – Y) il canale X è pari alla semisomma di A e B il canale Y è pari alla semidifferenza tra A e B.

Questa modalità è l'unica che consente un ascolto accettabile in mono di un canale originariamente stereo. La modalità tradizionale stereo, se ascoltata in mono, non farebbe altro che far ascoltare metà del patrimonio del brano.

Il joint stereo è consigliabile però soltanto a bassi bitrate; a bitrate alti, data la forte possibilità di errore, è sempre consigliabile la codifica stereo tradizionale che non è comunque obbligata a dare il 50% dei bit ad entrambi i canali ma può oscillare in base alle varie situazioni.

FILTRO PASSA-BASSO



Figura 2.1. Schema di un filtro passa-basso

Il filtro passa-basso è uno degli strumenti utilizzati per diminuire il bitrate del file MP3. Come precedentemente detto, le frequenze al di sopra dei 18 mila hertz non sono perfettamente udibili da parte del nostro apparato uditivo e quindi possono essere eliminate.

Il filtro passa-basso esegue esattamente questa operazione:

- a 128 kbps si usa un filtro passa-basso a 17500 hz;
- a 160 kbps si usa un filtro passa-basso a 18500 hz;
- a 192 e 224kbps si usa un filtro passa-basso a 19500 e 20000 hz;
- ai bitrate più alti e con il VBR alta qualità si utilizza un filtro passabasso a 20 khz.

1 FRAME MP3

Il frame è l'equivalente del fotogramma video e contiene una parte di suono. Normalmente, il frame dura qualche centesimo di secondo, portando il numero di frame al secondo a circa 40, indipendentemente dal bitrate. L'encoder può decidere di utilizzare frame a durata più bassa e, se è necessario, codificare meglio le informazioni contenute nel brano originario.

2.2. Come funziona WMA

WMA è il più comune codec audio della Microsoft. La sigla WMA, teoricamente, non si riferirebbe soltanto a questo codec ma commercialmente si fa un riferimento diretto ad esso. La prima versione di questo codec fu rilasciata nel 1999 come WMA 1. WMA è un codec a perdita di informazione basato sulla psicoacustica. Similarmente a quanto effettuato da altri codec come MP3, si cerca di eliminare le informazioni non udibili correttamente dall'uomo.

Fondamentalmente, la codifica WMA si basa sulla trasformata discreta del coseno modificata, in modo molto simile a quanto fatto dai codec AAC o Vorbis. Il flusso di WMA è composto da superframe, ciascuno contenente uno o più frame contenenti a loro volta 2048 campioni.

Ciascun frame contiene un numero dei blocchi lunghi 128, 256, 512, 1024, o 2048 campioni dopo la trasformazione nel dominio della frequenza tramite la trasformata. Nel dominio della freguenza, i campioni si mascherano e si riguantizzano. Successivamente, i campioni sono scomposti in due parti chiamate coefficiente ed esponente e codificati indipendentemente tramite la codifica di Huffman. Le informazioni stereo sono normalmente codificate in *middle/side*, che è poi la modalità joint. Ai bassi bitrate, le coppie di linee spettrali ed una forma di codifica del rumore possono essere utilizzate per migliorare la qualità di codifica. Come per AAC e Vorbis, WMA sfrutta le mancanze dello standard MP3. I tre formati hanno finito per essere disegnati in modo molto simile: sono ad esempio tre codec di pura trasformazione. L'implementazione della trasformata discreta del coseno modificata, usata nel codec WMA, è essenzialmente un superinsieme delle specifiche di Ogg e AAC, al punto che le routine di WMA possono essere usate per decodificare AAC e Vorbis in modo praticamente perfetto. Tuttavia, la quantizzazione e la codifica stereo dei tre codec sono molto diverse tra loro.

Un vero tratto distintivo del formato WMA è il suo originale uso di 5 differenti grandezze dei blocchi, a differenza di MP3, AAC e Vorbis che si limitano a due sole grandezze. WMA Pro estende le cinque grandezze portandole a 6 per le codifiche con una frequenza di campionamento compresa tra 88200 e 96000 hertz.

Anche WMA ha il suo formato di tagging per inserire all'interno del file informazioni aggiuntive su artista, titolo, album, ecc.

2.3. Come funziona Vorbis

Vorbis è un formato di codifica audio *open source* sviluppato principalmente da Christopher Montgomery, fondatore della Xiph.Org Foundation, che lo porta dalla prima alpha release del settembre 1998 al rilascio ufficiale della versione 1.0 avvenuto nel luglio 2002. Vorbis permette di campionare segnali da 8000 a 192000 volte al secondo e con una vasta gamma di modalità (monaurale, polifonia, stereo, quadrifonia, 5.1 e fino a 255 canali discreti).

Dato un segnale stereo campionato a 44100 hertz, l'encoder Vorbis produce un output che può andare da 45 a 500 kbit al secondo, in base alla qualità scelta. Vorbis concretizza un ragionamento diverso rispetto a molti altri codec fratelli. Essendo un algoritmo VBR, la Xiph.Org, piuttosto che considerare la qualità del segnale sulla base del numero di bit per secondo, raccomanda di considerare la qualità tramite la nomenclatura q. Il primo encoder aveva una gamma di qualità che andavano da q0 a q10, per numeri interi. Successivamente, è stato inserito un valore q - 1 corrispondente a circa 45 kbit per secondo.

Il codec Vorbis non è quindi in grado di garantire che un segnale sia, neanche mediamente, ad un numero preciso di bit per secondo, anche perché i pacchetti in cui è scomposto il flusso in ingresso non hanno né grandezza massima, né minima, né prefissata.

2.3.1. Dentro l'algoritmo

Vorbis è un codec monolitico e adattativo basato sulla trasformata discreta del coseno modificata. La trasformata viene utilizzata per mutare il suono dal dominio del tempo al dominio della frequenza. I dati nel dominio della frequenza sono scomposti in rumore di fondo e componenti residui, quindi quantizzati e codificati utilizzando un algoritmo apposito. La decompressione effettua gli stessi passaggi al contrario. L'approccio utilizzato dall'algoritmo non permette una buona qualità a bitrate bassi: si sente come un riverbero o un suono ovattato.

2.3.2. Il bitrate peeling

Il formato Vorbis supporta il *bitrate peeling* per ridurre il bitrate del file codificato. Il bitrate peeling è la capacità di un codec di utilizzare lo stesso file a bitrate più bassi senza decodificarlo e ricodificarlo nuovamente. Nonostante sia una possibilità del codec, la qualità è nettamente più bassa rispetto ad una ricodifica. Questa differenza di qualità nasce dal fatto che il codec non attribuisce un'importanza relativa ai dati, e quindi il peeling ha il non facile compito di capire cosa è importante e cosa no, senza potere ricodificare.

2.3.3. I container di Vorbis

Uno degli errori più frequenti parlando di Vorbis è collegarlo immediatamente ad Ogg. Ogg è un contenitore che può ospitare al suo interno diversi codec, come ad esempio nel caso di Ogg Theora. Similarmente, Vorbis può essere incapsulato in altri formati di contenitore (uno su tutti: Matroska) ma può anche essere il payload di un formato RTP (*real time protocol*).

2.3.4. I metadata

I metadata di Vorbis, chiamati "commenti", sono simili a i tag ID3 di MP3. I metadata sono memorizzati in un vettore di stringhe ad 8 bit, di lunghezza arbitraria. Il vettore è contenuto all'interno dell'header del secondo pacchetto che comincia un flusso Vorbis. Le stringhe sono codificate come UTF-8.

Le stringhe sono implementate nella forma [TAG]=[VALORE], ad esempio "ARTIST=La mia band". I tag sono *case-insensitive*.

Come nell'attuale versione di ID3, gli utenti e gli encoder possono inserire tag di qualsiasi tipo coerente con il flusso. Ad esempio, un encoder può utilizzare più tag per il genere.

2.4. Come funziona AAC

AAC è l'acronimo di *Advanced Audio Coding*. Creato dal consorzio MPEG, in origine veniva chiamato anche MPEG-2 NBC (*Not Backward Compatible*) in quanto non retro-compatibile con i formati già esistenti. Tra le caratteristiche principali, AAC:

- gestisce frequenze di campionamento da 8 a 96 kHz;
- gestisce bitrate da 8 a 512 Kbit/s;
- utilizza una codifica percettiva.

AAC funziona in modo molto simile ad MP3. I blocchi sono stati migliorati, aggiunte nuove funzioni in modo da rendere migliore la codifica a bassi bitrate.

AAC utilizza una codifica a tasso variabile; un file codificato con questo codec ha una qualità migliore rispetto a MP3, a parità di bitrate. Il codec non è dotato di metadati nativi: si utilizzano i tag ID3 di MP3.

2.4.1. Streaming

AAC nasce per lo streaming. Lo streaming audio è spesso veicolato dal protocollo di trasporto TCP; tuttavia, ritrasmettere dati non è conveniente per applicazioni audio, motivo per cui, a differenza di MP3 che implementa soltanto il CRC (controllo a ridondanza ciclica), AAC implementa diversi meccanismi di controllo degli errori:

- *error concealment*: CRC rileva gli errori che poi vengono corretti tramite la ricostruzione delle parti mancanti, con metodi di approssimazione o aggiunta di parti silenti;
- *error resilience*: irrobustisce il codec;
- error detection: diversi CRC consentono di rilevare errori;
- *error protection*: suddivide il flusso in varie parti e, applicando bit di controllo, consente di proteggere ulteriormente da errori sui bit.

2.5. Come funziona FLAC

Il codec FLAC è uno dei più importanti codec open source senza perdita utilizzati a livello mondiale.

FLAC permette di avere file compressi fino anche al 40, 50% rispetto al PCM originario, mantenendo tutti i dati del file originario, esattamente come avviene con i formati ZIP o RAR, che sono palesemente lossless.

2.5.1. Passaggi di funzionamento

SUDDIVISIONE IN BLOCCHI

L'input viene diviso in più parti contigue, anche variabili in grandezza.

COMPATTAMENTO DEL FLUSSO MULTICANALE

In questo *step* l'encoder FLAC si occupa di calcolare, nel caso di input stereo e surround, la media dei canali e la loro differenza. Il segnale a qualità migliore viene passato al processo successivo.

PREDIZIONE

Partendo dal primo blocco avviene la previsione di quale possa essere il successivo con degli algoritmi matematici che tentano di ricostruire il segnale. In questo caso verranno codificati solo i parametri, dato che gli algoritmi matematici devono essere presenti sia sull'encoder che sul decoder.

CODIFICA RESIDUA

La codifica residua permette a FLAC di essere effettivamente un codec lossless. Si codifica senza perdita tutta la parte di segnale che si differenzia dalla ricostruzione matematica di predizione e viene incorporata nel file finale. Il guadagno si ha sul fatto che il segnale residuo, proprio per la scrematura matematica, sarà minore in byte rispetto al corrispettivo PCM.

Capitolo 3 – Shoutcast

Shoutcast è un programma proprietario, prodotto dalla NullSoft (acquisita da AOL nel 1999) che consente lo streaming audio su Internet; si tratta, quindi, di un server di streaming, tra i più utilizzati al mondo anche per la sua disponibilità verso molti sistemi operativi (Windows, FreeBSD, Gnu/Linux tra questi). Per studiare meglio il software esamineremo due versioni dello stesso, in ambiente Windows. La prima versione esaminata è la 1.9; la seconda, la 2.0.

3.1. Versione 1.9

La versione 1.9 presenta una GUI (Graphic User Interface) che consente una rapida visione e un rapido accesso alle opzioni settabili.

La barra dei menu ha le seguenti opzioni:

- kill server: uccide il server, non rendendolo più funzionante;
- *hide monitor*: pone in background il server (sparisce dalla barra delle applicazioni, ma non viene chiuso);

- *edit log*: permette di controllare e modificare il file log, con tutto quello che è successo al server da quando è stato aperto;
- *edit config*: permette di settare le configurazioni del server. È, questa, la parte più importante da conoscere.

🕏 Nullsoft SHOUTcast Server M	or 🗙
Kjill server Hide Monitor Edit log Edit	g <u>A</u> bout
** SHOUT cast Distributed Network Juu Copyright (C) 1999-2004 Muldoft, Inc ** Use **sc_serv Bename n** to specify Event log (20/02/13/922214.32> [SHOUT cast] [(20/02/13/922214.32> [sman] loaded (20/02/13/92214.32> [sman] loaded (20/02/13/92214.32> [sman] loaded (20/02/13/92214.32> [sman] loaded (20/02/13/92214.32> [sman] loaded (20/02/13/92214.33> [sman] loaded (20/02/13/92214.33> [sman] loaded (20/02/13/92214.33> [sman] calert m (20/02/13/92214.33> [sman] calert m	vert ife
<	×
v1.9.7/win32	02/02/13 22:14:36 (0/32 (no source))

Figura 3.1. Finestra iniziale della GUI del server Shoutcast 1.9.7

Esaminiamo le righe che compaiono al centro della finestra:

```
[Shoutcast] DNAS/win32 v1.9.7 (Jun 23 2006) starting up...
[main] loaded config from C:\Programmi\Shoutcast\sc_serv.ini
```

Queste due righe indicano che è stato caricato con successo il file di configurazione.

```
[main] initializing (usermax:32 portbase:8000)...
```

Questa riga identifica che la radio può avere massimo 32 ascoltatori e occupa la porta 8000.

[main] No ban file found (sc_serv.ban)
[main] No rip file found (sc_serv.rip)

Queste due righe mostrano che non sono presenti i file ban e rip, due file utili per settaggi avanzati e non strettamente inerenti la trasmissione.

[main] opening source socket [main] source thread starting [main] opening client socket [source]listening for connection on port 8001 [main] Client Stream thread [0] starting [main] client main thread starting

Queste righe mostrano che il server si è posto in ricezione del flusso audio da inviare in streaming: a questo punto è pronto per effettuare la diffusione della radio che vi si collegherà.

3.1.1. Il file di configurazione

Cliccando su *edit config* si aprirà un editor di testo che consente di modificare le impostazioni del file *sc_serv.ini*, il file che fornisce le direttive al server Shoutcast.

Tutto il testo preceduto da un punto e virgola è un commento; il testo che non lo è rappresenta quindi una impostazione modificabile.

- ; MaxUser. The maximum number of simultaneous listeners allowed.
- ; Compute a reasonable value for your available upstream bandwidth (i.e. if
- ; you have 256kbps upload DSL, and want to broadcast at 24kbps, you would
- ; choose 256kbps/24kbps=10 maximum listeners.) Setting this value higher
- ; only wastes RAM and screws up your broadcast when more people connect
- ; than you can support.
- MaxUser=32

L'impostazione *MaxUser* permette di modificare il numero massimo di ascoltatori che può avere la vostra radio. Il numero va impostato in base ad alcuni parametri, tra cui la qualità audio con cui si trasmette e la banda in upload disponibile. Il numero deve essere intero e positivo.

; Password. While Shoutcast never asks a listener for a password, a

- ; password is required to broadcast through the server, and to perform
- ; administration via the web interface to this server. This server should
- ; consist of only letters and numbers, and is the same server your broadcaster
- ; will need to enter in the Shoutcast Source Plug-in for Winamp. THIS VALUE
- ; CANNOT BE BLANK.

Password=changeme

L'impostazione *password* permette di settare una password di sicurezza, che deve essere riproposta all'interno di un encoder che voglia connettersi al server. Non è obbligatorio cambiarla, anche se deve essere obbligatoriamente settata.

; PortBase. This is the IP port number your server will run on. The ; value, and the value + 1 must be available. If you get a fatal error when ; the DNAS is setting up a socket on startup, make sure nothing else on the ; machine is running on the same port (telnet localhost portnumber -- if you ; get connection refused then you're clear to use that port). Ports < 1024 are not allowed. **Portbase**=8000

L'impostazione *portbase* permette di modificare il numero di porta TCP alla quale il server Shoutcast risponderà. La porta 8000 è standard; non sono ammesse porte con numero inferiore a 1024. Con l'opzione *portbase* terminano le configurazioni fondamentali. All'interno dello stesso file cominciano le configurazioni opzionali.

; LogFile: file to use for logging. Can be '/dev/null' or 'none' ; or empty to turn off logging. The default is ./sc_serv.log ; on *nix systems or sc_serv_dir\sc_serv.log on win32. ; Note: on win32 systems if no path is specified the location is ; in the same dir as the executable, on *nix systems it is in the ; current directory. LogFile=sc_serv.log

L'impostazione *LogFile* permette di settare il nome del file contenente tutti le risposte del server ai vari eventi che possono verificarsi (ad esempio l'ingresso di un nuovo ascoltatore).

; RealTime displays a status line that is updated every second ; with the latest information on the current stream (*nix and win32 ; console systems only) RealTime=1

L'impostazione RealTime è funzionale ad un rapido aggiornamento del

server in caso di alto traffico di ascoltatori o di possibili eventi.

- ; ScreenLog controls whether logging is printed to the screen or not
- ; on *nix and win32 console systems. It is useful to disable this when
- ; running servers in background without their own terminals. Default is 1
- ScreenLog=1

L'opzione *ScreenLog* consente di stampare o non stampare a video il log del server, ma eventualmente solo su file. Utile per server che operano in background.

```
; ShowLastSongs specifies how many songs to list in the /played.html
; page. The default is 10. Acceptable entries are 1 to 20.
ShowLastSongs=10
```

All'interno dell'interfaccia web di Shoutcast è possibile visualizzare l'elenco delle ultime canzoni trasmesse. L'opzione *ShowLastSongs* indica quante di esse devono comparire in un particolare momento.

- ; TchLog decides whether or not the DNAS logfile should track yp
- ; directory touches. Adds and removes still appear regardless of
- ; this setting.
- ; Default is yes
- ; TchLog=yes

L'opzione *TchLog* permette di inserire all'interno dei log anche i contatti ricevuti periodicamente dalle directory delle *yellow pages*.

- ; WebLog decides whether or not hits to http:// on this DNAS will
- ; be logged. Most people leave this off because the DSP plug-in
- ; uses http:// calls to update titles and get the listener count,
- ; which takes up a lot of log space eventually. If you want to
- ; see people making hits on your admin.cgi or index pages, turn
- ; this back on. Note that this setting does NOT affect XML stats
- ; counters for hits to http:// pages.
- ; Default is no.
- ; WebLog=no

L'opzione *WebLog* permette invece di inserire all'interno del file di log anche i contatti ricevuti dall'interfaccia web, che possono però creare un quantitativo di dati ingente all'interno del file di log stesso.

; W3CEnable turns on W3C Logging. W3C logs contain httpd-like accounts
; of every track played for every listener, including byte counts those listeners
; took. This data can be parsed with tools like Analog and WebTrends, or given
; to third parties like Arbitron and Measurecast for their reporting systems.
; Default is Yes (enabled).

W3CEnable=Yes

L'opzione *W3CEnable* consente di tracciare il numero di byte ricevuti da ciascun ascoltatore, i brani che ha ascoltato ed altri dati che risulteranno poi trattabili per statistiche.

; W3CLog describes the name of the logfile for W3C logging. Default logfile is ; sc_w3c.log, in the same directory wherever the DNAS gets started from. W3CLog=sc_w3c.log

L'opzione *W3CLog* permette di dare un nome al file di log contenente i dati creati tramite l'opzione *W3CEnable*.

La terza sezione del file di configurazione riguarda i settaggi di rete, utili per la connettività del server con il resto del mondo.

; SrcIP, the interface to listen for source connections on (or to make relay

; connections on if relaying). Can and usually will be ANY or 127.0.0.1

; (Making it 127.0.0.1 will keep other machines from being able to

; broadcast using your Shoutcast server)

SrcIP=ANY

L'opzione *SrcIP* rappresenta la possibilità di un server Shoutcast di scegliere da quale IP possa essere usato. Inserire ANY consentirà al server di essere utilizzabile da tutti, cosa che succede impostando anche l'IP di loopback.

; DestIP, IP to listen for clients on (and to contact yp.Shoutcast.com)

; can and usually will be be ANY. If your machine has multiple IP addresses,

```
; set this to the one you want it to be accessed by. DestIP=ANY
```

L'opzione *DestIP* è utile esclusivamente per le macchine che possiedono più schede di rete, in quanto permette di scegliere l'IP al quale il server sarà disponibile.

; Yport, port to connect to yp.Shoutcast.com on. For people behind caching

- ; webproxies, change this to the alternate port (666 is what it might be,
- ; check www.Shoutcast.com if you have problems). Otherwise, leave this at 80.
- ; We're actively working on re-opening port 666, but as of release the only
- ; working port is port 80.
- Yport=80

L'opzione *Yport* indica al server attraverso quale porta contattare il sito yp.Shoutcast.com, che altro non è che il sito delle pagine gialle di Shoutcast.

- ; NameLookups. Specify 1 to perform reverse DNS on connections.
- ; This option may increase the time it takes to connect to your
- ; server if your DNS server is slow. Default is 0 (off).
- NameLookups=0

Un'opzione curiosa è la *NameLookups*, la quale consente di tentare un *reverse lookup* sulle connessioni in ingresso, cercando di ricavare dati su di essa. Può appesantire il server ed allungare i tempi di connessione in caso di traffico elevato.

; RelayPort and RelayServer specify that you want to be a relay server.

- ; Relay servers act as clients to another server, and rebroadcast.
- ; Set RelayPort to 0, RelayServer to empty, or just leave these commented
- ; out to disable relay mode.
- ; RelayPort=8000
- ; RelayServer=192.168.1.58

Se il server Shoutcast sul quale si sta lavorando deve essere adibito

a relay (ripetitore) di un altro server Shoutcast (ad esempio per aumentare il numero di ascoltatori potenzialmente disponibili), allora l'opzione *RelayPort* deve contenere la porta del server da ritrasmettere e l'opzione *RelayServer* il suo indirizzo IP. Dopo le opzioni di rete, la quarta sezione riguarda le opzioni di amministrazione.

; AdminPassword. This password (if specified) changes the

; behavior of Password to be a broadcast-only password, and

; limits HTTP administration tasks to the password specified

; here. The broadcaster, with the password above, can still

; log in and view connected users, but only the AdminPassword

; will grant the right to kick, ban, and specify reserve hosts.

; The default is undefined (Password allows control for both

; source and admin)

; AdminPassword=adminpass

L'opzione *AdminPassword*, se impostata, consente di aumentare il livello di sicurezza del server impedendo a chi non abbia questa password di potere effettuare il ban o di potere riservare uno slot ad un particolare indirizzo IP.

; AutoDumpUsers controls whether listeners are disconnected if the source ; stream disconnects. The default is 0.

AutoDumpUsers=0

L'opzione *AutoDumpUsers* permette di disconnettere dal server tutti gli ascoltatori, nel caso in cui la sorgente non fosse più in collegamento con il server.

; AutoDumpSourceTime specifies how long, in seconds, the source stream is

; allowed to be idle before the server disconnects it. 0 will let the source

; stream idle indefinately before disconnecting. The default is 30.

AutoDumpSourceTime=30

L'opzione *AutoDumpSourceTime* indica al server quanto tempo in secondi deve aspettare una sorgente inattiva prima di disconnetterla. Impostare 0 equivale ad indicare un valore infinito.

- ; ContentDir specifies the directory location on disk of where to stream
- ; on-demand content from. Subdirectories are supported as of DNAS 1.8.2.
- ; Default is ./content, meaning a directory named content in the same directory
- ; as where sc_serv was invoked from.
- ; ContentDir=./content

Shoutcast è in grado di fornire contenuti on demand; tali contenuti si devono trovare all'interno della cartella impostata dall'opzione *ContentDir.* I caratteri [./] indicano che la directory di base è quella dove si trova il server Shoutcast, ovunque essa sia.

- ; IntroFile can specify a MP3 file that will be streamed to listeners right
- ; when they connect before they hear the live stream.
- ; Note that the intro file MUST be the same samplerate/channels as the
- ; live stream in order for this to work properly. Although bitrate CAN
- ; vary, you can use '%d' to specify the bitrate in the filename
- ; (i.e. C:\intro%d.MP3 would be C:\intro64.MP3 if you are casting at 64kbps).
- ; The default is no IntroFile
- ; IntroFile=c:\intro%d.MP3

L'opzione *IntroFile* è molto utile per creare un jingle di presentazione dell'emittente da mandare all'avvio dell'ascolto della radio. Il file deve avere lo stesso bitrate e numero di canali dello stream attualmente trasmesso dalla sorgente (in ogni caso, mai superiore). Si possono impostare più file basati sul bitrate scrivendo [%d] che indica il bitrate dello stream.

- ; BackupFile can specify a MP3 file that will be streamed to listeners over
- ; and over again when the source stream disconnects. AutoDumpUsers must be
- ; 0 to use this feature. When the source stream reconnects, the listeners
- ; are rejoined into the live broadcast.
- ; Note that the backup file MUST be the same samplerate/channels as the
- ; live stream in order for this to work properly. Although bitrate CAN
- ; vary, you can use '%d' to specify the bitrate in the filename

; (i.e. C:\backup%d.MP3 would be C:\backup32.MP3 if you are casting at 32kbps).

- ; The default is no BackupFile
- ; BackupFile=C:\intro%d.MP3

Un'opzione alternativa all'*AutoDumpUsers* (che in questo caso deve essere impostata a zero) è la funzione *Backup*, che consente, quando una sorgente non è più attiva, di trasmettere continuamente un singolo file (che deve sottostare alle stesse regole viste per l'opzione *IntroFile*).

; TitleFormat specifies a format string for what title is sent to the listener.

; For example, a string of 'Justin Radio' forces the title 'Justin Radio' even

; when the source changes the title. You can use up to one '%s' in the string

; which lets you contain the title from the source. For example, if your

; TitleFormat is 'Justin Radio: %s', and the source plug-in's title is

; 'Billy plays the blues', then the net title is

; 'Justin Radio: Billy plays the blues'. Note: only works on non-relay servers.

; The default is no format string.

; TitleFormat=Justin Radio: %s

L'opzione *TitleFormat* consente di visualizzare, all'interno dei player degli ascoltatori, una stringa organizzata come meglio si reputi; ad esempio, inserendo sempre il nome della radio piuttosto che i titoli che variano canzone dopo canzone, oppure un ibrido tra le due cose, come nell'esempio di default. [%s] indica il titolo del brano corrente. Funziona solo con server non in modalità relay.

; URLFormat specifies a format string for what URL is sent to the listener.

; Behaves like TitleFormat (see above).

; The default is no format string.

; URLFormat=http://www.server.com/redirect.cgi?URL=%s

L'opzione *URLFormat* consente di inviare all'ascoltatore un URL che sia corrispondente, ad esempio, al titolo del brano corrente, ed ha lo stesso comportamento di *TitleFormat*. L'opzione non è abilitata di default.

; PublicServer can be always, never, or default (the default, heh)

; Any setting other than default will override the public status

; of the source plug-in or of a Shoutcast server that is being relayed.

PublicServer=default

L'opzione *PublicServer* può forzare il server ad essere pubblico, privato