

動画・音声コンテンツ 伝送技術のすべて

IC Series
Industrial
Computing

マルチメディア・ ストリーミング技術

笠野 英松◎著

見本

- マルチメディア・ネットワーク・システムを支える基礎理論と標準規格の解説、ストリーミング・サーバの運用と実装、サーバ動作の分析、パソコンやBREW/Androidクライアントへの実装方法、コンテンツ産業と技術動向までを網羅。

```
/** MPEG-4/MP3 インスタンス作成 **/  
(IMedia *)pMe->pMedia = NULL; //default  
if ( mp4sw ) //MP4 ファイル / ストリーミング再生の場合  
    ret = ISHELL CreateInstance( (IShell *)pMe->pIShell, AECLSID_MEDIAMPEG4, (void **) &pMe->pMedia);  
else //MP3 ファイル再生の場合  
    ret = ISHELL CreateInstance( (IShell *)pMe->pIShell, AECLSID_MEDIAMP3, (void **) &pMe->pMedia);  
// 実行結果のチェック  
switch (ret) {  
    case SUCCESS: // クラスが作成された  
        break;  
    case ENOMEMORY:  
        CHPrintf( pMe, "CLSID: ENOMEMORY" );  
        return FALSE;  
    case ECLASSNOTSUPPORT: // 指定したクラスが未サポート  
        CHPrintf( pMe, "CLSID: ECLASSNOTSUPPORT" );  
        return FALSE;  
    default: // その他  
        CHPrintf( pMe, "CLSID: Other" );  
        return FALSE;  
}  
  
/** MPEG4 情報 (画像&組み込み文字) 描画区画の作成. 文字領域: 画面上 10分の1, 画像領域: その下に画面高さの10分の7 ***/  
SETAERECT( &pMe->prImage, 0, pMe->DeviceInfo.cyScreen / 10, pMe->DeviceInfo.cxScreen, 7 * pMe->DeviceInfo.cyScreen / 10 );  
SETAERECT( &pMe->prText, 0, 0, pMe->DeviceInfo.cxScreen, pMe->DeviceInfo.cyScreen / 10 );  
IMEDIA_SetRect( pMe->pMedia, &pMe->prImage, &pMe->prText );  
/**/ スティア・データの設定とレディ状態への移行 ***/  
ret = IMEDIA_SetMediaData( pMe->pMedia, &pMe->pad );  
// 実行結果のチェック  
switch (ret) {  
    case SUCCESS: // 成功  
        break;  
    case EBADPARAM:  
        CHPrintf( pMe, "SetMedia: EBADPARAM" );  
        return FALSE;  
    case EBADSTATE: // 取得不可状態  
        CHPrintf( pMe, "SetMedia: EBADSTATE" );  
        return FALSE;  
    case EUNSUPPORTED: // 未サポート  
        CHPrintf( pMe, "SetMedia: EUNSUPPORTED" );  
        return FALSE;  
    default: // その他  
        CHPrintf( pMe, "SetMedia: Other" );
```

CQ出版社

目次

はじめに	3
第1章 基礎理論	9
1.1 マルチメディア伝送再生方式	10
1.1.1 ダウンロード方式	10
1.1.2 プログレッシブ・ダウンロード方式	10
1.1.3 リアルタイム・ストリーミング方式	11
1.1.4 擬似ストリーミング伝送再生の拡張	12
1.2 符号化処理技術	12
1.2.1 変換, 圧縮, 量子化	12
1.2.2 量子化	14
1.2.3 画像処理における主要な符号化技術	18
1.2.4 音声符号化技術のベース	22
1.3 マルチメディア・ネットワーク	24
第2章 マルチメディア伝送基盤の技術標準	25
2.1 光ファイバ網	26
2.1.1 高速デジタル伝送技術	27
2.2 無線	31
2.3 多重化と多元化	39
2.4 バックボーン技術(WDM/DWDM)	41
2.5 有線LAN/MAN—IEEE802.3規格	42
2.5.1 階層構造	42
2.5.2 フレーム構造	44
2.5.3 Ethernet標準の概要	45
2.6 無線ネットワーク	50
2.6.1 WPAN(IEEE802.15)	50
2.6.2 無線LAN—IEEE802.11	53
2.7 無線MAN(WAN)—IEEE802.16	55
2.8 モバイル広帯域無線アクセス(IEEE802.20)	59
2.9 マルチメディア伝送プロトコル	63

第3章 マルチメディア標準規格	71
3.1 マルチメディア標準規格と符号化技術の概要.....	72
3.1.1 画像符号化の歴史.....	72
3.1.2 画像処理の主要な符号化技術.....	74
3.1.3 音声符号化技術.....	74
3.1.4 MPEG オーディオ符号化.....	75
3.1.5 MPEG オーディオ符号化の拡張.....	76
3.2 静止画像圧縮符号化.....	78
3.2.1 JPEG/T.81.....	78
3.2.2 JBIG/T.82.....	79
3.2.3 モーションJPEG.....	79
3.2.4 GIF.....	79
3.2.5 PNG.....	80
3.2.6 TIFF.....	81
3.2.7 PICT, BMP.....	81
3.3 動画画像圧縮符号化.....	81
3.3.1 MPEG-1/MPEG-2.....	82
3.3.2 H.261/H.263/H.264.....	84
3.3.3 MPEG-4.....	86
3.3.4 MPEG-7.....	95
3.3.5 MPEG-21.....	98
3.3.6 3GPP マルチメディア・ファイル・フォーマット.....	98
3.3.7 3GPP2 マルチメディア・ファイル・フォーマット.....	98
3.4 通信処理.....	99
3.4.1 多重化 (H.221/H.222.0=MPEG-2TS/H.223).....	99
3.4.2 通信制御 (H.242/H.245).....	102
3.4.3 マルチメディア通信システム (H.233/H.234/H.310/H.320/H.321/H.323/H.324, T.12x).....	103
3.5 ハイパー処理.....	106
3.5.1 MHEG.....	106
3.5.2 SGML/HTML.....	107
3.5.3 XML/XHTML/VRML/SOAP.....	107
3.6 ストリーミング.....	108
3.6.1 リアルタイム・ストリーミングの原理.....	108

目次

3.6.2	ストリーミング製品	109
3.6.3	ストリーミングのポイント	111
3.7	映像信号フォーマット	112
3.8	そのほかの符号化	113
第4章 マルチメディア・ネットワーク・システムの利用と運用115		
4.1	コンテンツの作成	116
4.1.1	エンコード	116
4.1.2	RealProducer	117
4.1.3	Microsoft Expression Encoder	119
4.1.4	MPEG-4規格のエンコード	120
4.2	ストリーミング・サーバ	122
4.2.1	QuickTime Streaming Server (Darwin Streaming Server)	122
4.2.2	Helix サーバ	133
4.2.3	Windows Media Server	145
4.2.4	オープン・ソースのストリーミング・サーバ	155
4.2.5	ライブ・ストリーミング	155
4.3	クライアント側で対応するメディア	167
4.4	擬似ストリーミング	168
コラム Windows Media Playerでのユーザ名/パスワード入力のリフレッシュ方法.....154		
第5章 マルチメディア・ネットワーク・システムの実装169		
5.1	ストリーミングの手順	170
5.2	ストリーミング・サーバ	175
5.3	ストリーミングの実装	189
5.3.1	VODストリーミング(testOnDemandRTSPServer)	190
5.3.2	マルチキャスト・ストリーミング(testMPEG1or2AudioVideoStreamer)	194
5.3.3	vlc	196
5.3.4	RFC3550	205
5.4	Windowsクライアント・ストリーミング・プログラミング	206
5.4.1	ストリーミング再生プログラミングの手順	208
5.4.2	QuickTime APIによるWindowsクライアント・ストリーミング・プログラム	211

5.5	そのほかの実装	212
第6章	モバイル・クライアントやストリーミング関連の実装	213
6.1	BREWクライアント・ストリーミング・プログラム	214
6.2	Androidクライアント・ストリーミング・プログラム	230
6.3	システム負荷分散制御	233
6.4	近似ストリーミング	234
6.5	そのほかの実装	237
第7章	モニタ分析	239
7.1	分析用中継システムの作成	240
7.1.1	マルチスレッド処理	242
7.1.2	RTSPストリーミングの必須および推奨メソッドに限定した処理	243
7.1.3	処理中の記録はメモリ上に保存，処理完了後はディスクに書き込む	244
7.1.4	収集する記録データ	244
7.1.5	クローズ処理	245
7.2	中継ログによるデータ分析	245
7.2.1	分析の対象と利用するデータ	245
7.2.2	中継システム経由のRTSPストリーミング	251
7.2.3	中継システムによるポート転送	254
7.2.4	RTSPストリーミングでのパケット送出間隔	258
7.2.5	RTSP中継ストリーミングでのTCP/RTSP終結シーケンス	259
7.2.6	DSSのアクセス・ログ	261
第8章	マルチメディア・ネットワークの産業構造と新しい技術の流れ	263
8.1	マルチメディア・ネットワーク・システムの現状と動向	264
8.1.1	産業統計	264
8.1.2	マルチメディア伝送再生方式の新しい流れ	265
8.2	サービスにおける考察	268
8.2.1	サービスの保護	268
8.2.2	サービスのポイント	270



目次

Appendix1	QuickTime APIによるWindows クライアント・ストリーミング・プログラム	272
Appendix2	本書でのAndroid開発環境の構築と利用	280
Appendix3	マルチメディア・ネットワーク・サービスの新たな試み	284
	索引	289
	参考文献	293
	著者略歴	295

マルチメディア・ネットワーク・サービスは、現在では社会のインフラとして必要不可欠なものとなっています。その仕組みを理解するためには、マルチメディア・ネットワーク・システムにおけるデータの流れをはじめとして、伝送再生方式や符号化などの仕組み、そして、通信システムの機能の構成などについても理解しておく必要があります。本章ではそれらの基礎となる事柄について説明します。

1.1 マルチメディア伝送再生方式

システムとしてのマルチメディア・ネットワークには、コンテンツの作成、伝送/中継/再生、運用の三つの機能が必要です。さらに、運用においてはコンテンツのオンライン自動更新や、更新のための制御と管理、受信制御、変換などの機能も必要です。本書では解説していませんが、さまざまなWebコンテンツの作成はオーサリング・ツールなどを利用して行われるので、それらを活用するノウハウも高度なWebサイトの構築に関係する大きな分野です。そして、それらはサービスのデータ品質に大きく関わってきます。

マルチメディアを伝送・再生する方式には、ダウンロード方式、プログレッシブ・ダウンロード方式(擬似ストリーミング方式とも呼ばれる)、リアルタイム・ストリーミング方式という三つの方式があります。なお、コンテンツの配信を行う際には、著作権管理への十分な配慮と、著作権保護の技術的な仕組みが必須です。

1.1.1 ダウンロード方式

ダウンロード方式は、HTTPプロトコルでホームページ上のマルチメディア・コンテンツをダウンロードしてから、クライアント・システム上のプラグインで再生する、という最も単純な方式です。ただし、この方式は、コンテンツの全てのデータのダウンロードが完了するまで再生を開始できません。メリットは、標準のHTTPプロトコルによるクライアント/サーバ通信とメディア再生プラグイン(クライアント側の一般のマルチメディア再生プレイヤー)で簡単に利用できることです。

1.1.2 プログレッシブ・ダウンロード方式

ダウンロード方式の変形として、コンテンツのダウンロードが完了する前に、再生を開始できる一定量のデータをダウンロードして、そこから再生を開始する方式もありま

す。それが、プログレッシブ(段階的再生)ダウンロード方式です。

この方式もHTTPプロトコルなどを使用しますが、早く再生を開始できるので、「ファスト・スタート」方式とも呼ばれます。再生に必要なフレームなどの情報がファイルの先頭にあるため、その情報を使用して、必要なデータさえバッファリングされていれば再生が開始できます。

この方式は、後述するリアルタイム・ストリーミング方式と同様に、データ送受信と再生を同時に行うため、「擬似ストリーミング」方式と呼ばれることもあります。擬似ストリーミング方式の特徴として、RTP/RTSPではなくHTTPやFTPなどのデータ転送プロトコルを使用するためストリーミング専用サーバが不要で、データの保存が可能になるなどの長所があります。しかし、ライブ(生)放送が不可能で、頭出しなどのトリック・プレイができない、などの短所もあります。

なお、この方式で配信を行うには、一般のコンテンツからファスト・スタート用に変換保存したものでなければなりません。また、クライアント側の再生プレイヤーでは、このプログレッシブ・ダウンロード方式に対応した機能が必要です。

1.1.3 リアルタイム・ストリーミング方式

リアルタイム・ストリーミング方式は、擬似ストリーミング方式と同様にデータを受信しながら再生を行うストリーミング方式の一つです。特にストリーミング・プロトコル(RTP/RTSP)を使用して、マルチメディア送受信システムが一体となった時間軸の中で同期を取りながらデータの配信と再生を行います。このとき、クライアント側では受信データをバッファリングしますが、データの保存は行いません。

ストリーミング方式では、その配信ストリームの時間軸がライブなのかどうかで、ライブ配信またはオンデマンド配信の二つから選択することになります。ライブ配信は、複数のクライアントに同報配信し、全クライアントが同じ時間軸のコンテンツを視聴する形態(テレビやラジオなどの放送と同じ)です。オンデマンド配信は、リクエストに応じて各クライアントに対し個別に視聴させる形態です。

ライブ配信は、配信するイベントの進行中にそのメディア・ソースをリアルタイムにエンコードし、クライアントに配信します。これには、ブロードキャスト(一斉同報)ソフトウェアが使用されます。エンコード保存済みのビデオ・コンテンツも配信できます。

エンコード保存されたビデオ・コンテンツをオンデマンド配信する場合は、クライアントのリクエストに応じてユニキャスト(1対1対向)でストリーミング配信するのでブロードキャスト・ソフトウェアは不要です。

なお、ストリーミング方式で使用するコンテンツもストリーミング用に変換されていなければなりません。クライアント側のマルチメディア再生プレイヤーには、ストリーミング・プロトコル(RTP/RTSP)機能が必要です。

■ 3GPP2(UMB)

3GPP2の次世代UMB(Ultra Mobile Broadband；超モバイル広帯域)無線インターフェースは、以下のような仕様です。

- 前方向リンク(下り)：最大288Mbps, 逆方向リンク(上り)：最大75Mbps
- CDMAとOFDMAをシングル・インターフェース上にのせる
- MIMOとSDMA(Space Division Multiple Access；空間分割多重アクセス)
- 5MHz帯域幅あたり250を越えるVoIPユーザ・サポート
- LDPC：Low Density Parity(bit)Coding；低密度パリティ符号化
- 1.25～20MHzチャンネル帯域幅

2.9 マルチメディア伝送プロトコル

TCP/IPネットワークでの主要なマルチメディア伝送プロトコルには、RTP(Real-Time Transport Protocol), RTCP(RTP Control Protocol), ST2(Internet Stream Protocol Version 2), RTSP(Real-Time Streaming Protocol)などがあります。いずれもインターネットの技術文書RFCで標準化されています。

なお、マルチメディア・ストリーミング関連のプロトコル技術は、大きく分けるとストリーミング伝送、マルチキャスト、接続制御に分類され、これらは純然たるストリーミング伝送のプロトコルです。そのほかのマルチキャストでは、IGMP(Internet Group Management Protocol)をベースとするルーティング、接続制御ではSIP(Session Initiation Protocol)やSCTP(Stream Control Transmission Protocol), Megaco(Media Gateway Control), そのほかにリソース予約のRSVP(Resource Reservation Protocol)などがあります。

ここではストリーミング伝送、マルチキャスト、接続制御について解説します。

■ RTP(Real-time Transport Protocol - Version 2)

RTPはオーディオやビデオなどをリアルタイムにデータ伝送を行うプロトコルで、RTCPとともにRFC3550(RTP：A Transport Protocol for Real-Time Applications)とRFC3551(RTP Profile for Audio and Video Conferences with Minimal Control)で規定されています。RTPの特徴は、

- リソースの確保やQOS(品質)の保証などを行わない
- トランスポート層やネットワーク層のプロトコルを選ばない

という点にあります。とは言っても、最小限の送達確認や監視などを行うための制御プ

3.4 通信処理

3

マルチメディア情報の通信処理を図3.8に示します。ネットワーク伝送プロトコル上で、マルチメディア多重化・同期化処理により、その上層の音声や映像、データを多重化します。この処理を制御するシステム(通信)制御機能があり、呼制御はITU-TのQシリーズで行われます。こうしたマルチメディア伝送処理機能全体が、マルチメディア通信システムです。

本節では、このマルチメディア多重・同期、通信制御、システムを見ていきます。

3.4.1 多重化(H.221/H.222.0=MPEG-2TS/H.223)

マルチメディア通信では、音声や映像などの各種情報をリアルタイムに送受信側で同期をとりながら送受信するため、多重化して送受信しなければなりません。そのための同期/多重化方式が必要となります。この同期/多重化には、ビット単位で処理するビット同期/多重化方式と、各情報を個々にひとまとめのパケットにして処理するパケット同期/多重化方式の二つがあります。

ビット同期/多重化方式には、狭帯域ISDNのためのITU-Tのオーディオ・ビデオ伝送規格のH.221があります。

パケット同期/多重化方式には、MPEG-2のTS(Transport Stream, トランスポート・ストリーム)による伝送方式と、一般電話用のテレビ会議伝送規格であるITU-TのH.223規格があります。なお、ISOのMPEG-2はITU-TのH.262と共通文書であり、MPEG-2のTSもITU-TのH.222と共通文書です。

■ H.221

H.221^{※14}は、64kbps～1920kbpsまでの狭帯域ISDNを利用したマルチメディア通信

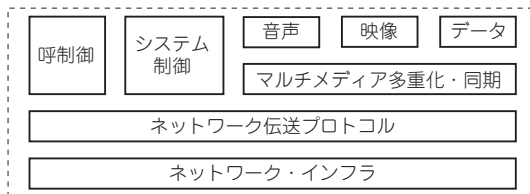


図3.8
マルチメディア通信処理

※14 H.221 : Frame structure for a 64 to 1920 kbit/s channel in audiovisual teleservices ; オーディオ・ビジュアル・テレサービスにおける64～1920kbpsチャネルのフレーム構成

(Remote Procedure Call, リモート手続き呼び出し)を実行することができます。つまり、HTMLなどの記述言語よりも文書を利用した送受信プログラムに近いものです。

3.6 ストリーミング

ブロードバンドの最大のターゲットは、マルチメディア・ストリーミングです。特に映像のストリーミングはさまざまな市場があると言われています。ストリーミングの基本的なプロトコルは、第2章の「マルチメディア伝送プロトコル」で解説したRTP/RTSPです。リアルタイム伝送を使用して送受信システムが一体となってストリーミングの同期再生を行います。典型的なアプリケーションが映像配信やライブ中継などです(図3.12)。

3.6.1 リアルタイム・ストリーミングの原理

現在のビデオ・ストリーミングは、リアルタイム・ストリーミング方式です。一般にTCP/IPのRTSP/RTPプロトコルを使って行われます。このRTSP、RTPはビデオ情報の送受信を制御するプロトコルと実際のビデオ・データを送受信するプロトコルです。送受信制御では、図3.13のようにストリーミング・データに関する情報やその通信に使われるポート(通路)などのネゴシエーション(調整)を文字形式で行います。一般に問題となる画像表示の開始が遅い原因の一つは、このネゴシエーション時間が大きいことです。

ネゴシエーションが完了しても、すぐに画像表示が開始されるわけではなく、クライアントではサーバからのビデオ・データを一旦、連続再生できる程度ためて(バッファリング)から、やっと表示が開始されます。概して、5秒~10秒という開始遅延は、こ

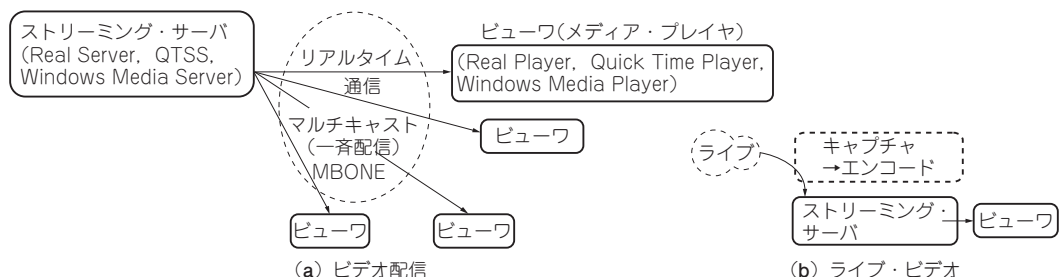


図3.12 ストリーミングの典型的な利用例

ブ・ストリーミングがあります。ここでは、ライブ・ストリーミングを解説します。

■ オンデマンド・ストリーミングとライブ・ストリーミング

オンデマンド・ストリーミングでは、ストリーミング・サーバにエンコードされた保存済みのマルチメディア・コンテンツを使用しますが、ライブ・ストリーミングではビデオやオーディオをリアルタイムに取得しなければなりません。そのため、ライブ・ビデオとライブ・オーディオをストリーミングするためのビデオ・カメラやマイク、ミキサなどの機器も必要です。ビデオやオーディオのソースには、USB Webcam, DV (Digital Video)カメラ (DV オーディオ, ビデオ FireWire/(IEEE 1394 出力), iSight カメラ, FireWire 対応アナログ DV コンバータ, FireWire DV オーディオ, USB オーディオ, 内蔵オーディオなどがあります。

オンデマンド・ストリーミングでは、クリップへのアクセスの最初はいつもその先頭から開始され、早送り、一時停止、巻き戻しなどのトリック操作を個別に行えますが、ライブ・ストリーミングでは、映像音声はリアルタイムなので、こうしたトリック操作ができません。また、入力デバイスから取得・変換したマルチメディア・データのエンコーダの出力先はファイルではなく、ローカルやリモートのストリーミング・サーバとします。そのサーバは、エンコードされたマルチメディア送信を受け付けてクライアントに配信するように設定しておきます。

ビデオ/オーディオをカメラ/マイクからリアルタイムに取得し、エンコードして出力するエンコーダのソフトウェアやシステムであるプロデューサが、直接クライアントからのリクエストを受け付けるライブ・ストリーミング・サーバになる場合もあります。

■ プッシュ型とプル型のライブ・ストリーミング

今までのライブ・ストリーミングではカメラ/マイクからリアルタイムに取得され、エンコーダでエンコードされたマルチメディア・ストリームが配信サーバに送られて、それをクライアントから視聴する方法が採られていました。しかし、これではストリーミング配信が必要ないときにもネットワーク上を無駄なデータが流れます。そこで、こうしたリクエストがない場合のために、プロデューサ側にクライアントからのリクエストに応じてストリーミング配信サーバが接続されてストリームを受け取る方式も採られています。プロデューサから送信する方式を「プッシュ型ライブ・ストリーミング」、そして、ストリーミング配信サーバから接続する方式を「プル型ライブ・ストリーミング」と呼んでいます。

Helix のエンコーダ (RealProducer, Helix Producer) や Microsoft のエンコーダ (Windows Media Encoder, Expression Encoder) では、これらプッシュ型およびプル型のライブ・ストリーミング配信をサポートしています。Apple 社のエンコーダ QuickTime Pro はライブ・ストリーミング配信に対応せず、QuickTimeBroadcaster (Mac OS X) がプッシュ型ライブ・ストリーミング配信に対応しています。

5.1 ストリーミングの手順

RTSP(Real Time Streaming Protocol)/RTP(Real-time Transport Protocol)ストリーミングは、一般的にRTSPやHTTPによりストリーミング情報の制御を行い、UDP(User Datagram Protocol)/RTPによりデータ・ストリーミングを行います。本章では、RTSPストリーミングと関連するユニキャストやマルチキャスト通信の手順を解説します。ほかの選択肢として擬似ストリーミングについて、第6章で実装の簡単な近似ストリーミングを解説します。

■ RTSPストリーミングの手順

ストリーミングで使用される再生・制御プロトコルを表5.1に示します。一般にユニキャストでは、単一ホストがストリーミング接続・通信する場合にRTSPを使用し、SDP(Session Description Protocol)セッション記述を動的に作成したり、ストリーミン

表5.1 ストリーミングのプロトコル

ストリーミング再生・制御プロトコル	概要
HTTP	HTTPによる制御
RTSP over TCP	一般のストリーミング制御。RFC仕様のRTSPではTCPとUDPを使用できるがTCPが一般的である
RTSP over UDP	Amino(http://amino.sourceforge.net/)/MPEG-2 TSで使用
RTSP over RTP(over UDP)	Windows Media PlayerのRTSP手法(つまり、通常のRTSPサーバとは通信できない)。データ・ソケット(RTP)と制御ソケット(RTSP)を使用。MPEG-4転送で、これを使用するカメラがある。サーバは554/TCP & UDPで転送メディア(イメージやオーディオなど)を問い合わせるのでUDPで処理できる
RTP over UDP	一般のストリーミングのデータ転送。RTCPをセッション制御にする。ユニキャストとマルチキャストがある。前者は最も効率的な方法かつ待ち時間が最短。後者は一般のネットワークではむだなトラフィックをネットワーク上に送出してしまう(Multicast Backboneネットワーク-MBONEで利用)
RTP over RTSP ^{注1}	ファイアウォールを経由するRTP処理用で、RTSP通過設定が簡単なことを利用。Live555 RTSP ライブラリ
RTSP/RTP over HTTP ^{注2}	HTTPトンネルにより、データと制御の二つを一つのソケットで処理。HTTP経由でトンネル化してファイアウォールを通過させる

注1 別名：RTP over TCP, interleaved RTP, RTSP/RTP interleaved mode, RTP tunneled over RTSP

注2 別名：RTSP/RTP tunneling mode, RTP over RTSP over HTTP

参考 RTPストリームをファイアウォールを通過させるためにRTSPプロキシ実装キットがある。
<http://sourceforge.net/projects/rtsp/>, http://sourceforge.jp/projects/sfnet_rtsp/(日本語版)

6.1 BREWクライアント・ストリーミング・プログラム

モバイル機器でもストリーミングは使われています。本章では、さまざまなプラットフォームのうち、BREWとAndroidでのストリーミングの実装方法を紹介します。Qualcomm社が提供するBREW^{※1}には、旧来のBREWに加えてオープンOSのBREW MP (Mobile Platform)があります。本節では、これら二つのSDKによるストリーミング・プログラムを紹介します^{※2}。

BREWがサポートするビデオ規格ではH.264/QCIFが、オーディオ規格ではQCPがそれぞれ標準です。RTSPストリーミングでは、コンテナ・フォーマット3gp/3gp2/mp4、ビデオ・コーデックMPEG-4/H.263/H.264、オーディオ・コーデックAAC/AMR-NB/AMR-WBに対応します。ただし、シミュレータではPCMオーディオ・ストリーミングのみに対応するので、ビデオ・ストリーミングでは音声が出ません。

BREW実機で使用するコンテンツの作成には、QuickTime player Proがよく使用されています。

■ BREWプログラミングの特徴

BREWのプログラミング環境は、WindowsのVisual C++環境でC言語を中心に作成していきます。そして、Visual C++の中でBREWアプリケーション・ウィザードを使用してBREWアプリケーションのフレーム(アプレット本体のフレーム、アプレット構造体、生成ルーチン、初期化ルーチン、解放ルーチン、イベント・ハンドラなど)を作成しておき、ユーザがその中に自らのコードを追加することになります。この手法は、Windowsの一般のVisual C++開発におけるMFCアプリケーションの作り方によく似ています。

また、実機でのプログラムmodファイルを作成する手順は、UnixにおけるCCやGCCによる手法によく似ています。

ただし、WindowsやUnixなどと異なる以下のようなBREW特有のポイントがあるので、それらに注意してプログラムを作成する必要があります。

※1 BREW (Binary Runtime Environment for Wireless) MP (Mobile Platform), <http://www.brewmp.com/>

※2 関係するURL

Brew MP ホーム (旧BREWを含む), <http://www.brewmp.com/>

旧BREW SDK 3.1 ホーム, <https://brewx.qualcomm.com/brew/sdk/download.jsp?page=dx/en/brew31/ad/t1/overview>

About Brew MP SDK, <https://developer.brewmp.com/content/about-brew-mp-sdk>

Download Brew MP SDK, <https://developer.brewmp.com/tools/brew-mp-sdk>

7.1 分析用中継システムの作成

本章では、ストリーミング・サーバとクライアントとの間のストリーミング・システムの動作を、実際のパケット・フローや手順、データ内容、タイム・スタンプ、アクセス・ログなどから見ていきます。

ストリーミング・サーバにはDSS(Darwin Streaming Server, バージョン6.03)を使用し、クライアントにはQuickTime Player(バージョン7.6.9)を使用します。そして、両者の間に置いた中継システムのログとtcpdump, クライアントのネットワーク・プロトコル・アナライザと、DSSのログにより分析を行います。中継システムとしてUNIX/Linux上で稼働するソフトウェアを、ネットワーク・プロトコル・アナライザ・ソフトウェアとしてWindows上のWireshark^{※1}を使用します。システム構成を図7.1に示します。

中継ソフトウェアinterserver 6(本書のサポート・サイトからダウンロード)は、UNIX/Linux上で動作するマルチスレッド処理プログラムです。プログラム作成時にPOSIXスレッド・ライブラリを使用するための「-pthread」^{※2}オプションを指定し、プログラム・オプションとしては一般的なデバッグ情報を取る「-DDEBUG」とUDPストリーミング・パケットを記録する「-DUDPPACKETINFO」の二つを指定します。

また、プログラム実行時に使用する設定ファイルとして、「intcont.tab2」があります。DSSのサーバ名とRTSPポート番号を記述したものです。RTSPポート番号はこの中継サーバがクライアントからのRTSP着信待ちをするポートでもあります。

```
c2g.example.com 554 # RTSP1(コメント)
```

なお、RTSPポート番号は複数記述(複数の中継サーバとして動作)が可能です。現実的には、システムの性能面に依存します。

プログラム処理で主要なポイントを以下に示します。

- ① マルチスレッド処理
- ② RTSPストリーミングの必須および推奨メソッドに限定した処理
- ③ 処理中の記録はメモリ上に保存し、処理完了後にディスクに書き込む
- ④ さまざまな記録データ

本プログラムは、図7.2のようなQuickTime PlayerとDSSとの間のRTSPストリーミングの典型的なRTSPシーケンスに特化して中継・分析するための作りになっていま

※1 Wireshark, <http://www.wireshark.org/>

※2 pthreads: POSIXスレッド。実際には、CentOS 5.5の場合、NPTL(Native POSIX Threads Library)2.5。

テンツ市場全体の12.5%)で、映像系コンテンツは4,251億円(29%)、音声系コンテンツは3,250億円(22%)です(そのほかは、テキスト系コンテンツ)。

平成21年のモバイル・コンテンツ産業の市場規模^{※7}は、対前年比12.4%増の1兆5,206億円で拡大しています。内訳は、携帯インターネットによるデジタル・コンテンツ取引の市場は5,525億円(対前年比14.3%増)、それ以外の物販・サービス・トランザクション系のモバイル・コマース市場は9,681億円(同11.4%増)です。

この概観からも、モバイル・ネットワーク化が進み、日本の産業の主役となっていることがよくわかります。

8.1.2 マルチメディア伝送再生方式の新しい流れ

RTSP/RTPストリーミングの特徴については良い面ばかりが強調されてきましたが、それはダウンロード方式やプログレッシブ・ダウンロード方式に対するものでした。しかし、標準的なプロトコルであるHTTPやFTPなどを使用すれば、通常のHTTPサーバ(FTPサーバ)を利用でき、サービス・アプリケーション自体もWWWアプリケーションあるいはWWWサービスとして、CGI/SSIにより簡単に拡張できます。そのためコスト面でも、クライアントやサーバにとって好都合になります。

リアルタイムRTSP(HTTPを含む)/RTPストリーミング以外のストリーミング手法のネックは、ライブ・ストリーミングに向かないという点が大きなものでした。しかし、第4章と第5章で解説したような近似ストリーミングという発想により、別の方向へと進んできています。

ここでは、そうした流れのベースとなる「HTTPプログレッシブ・ダウンロード・ストリーミング」について最初に解説し、新しい流れとしてApple社の「HTTP Live Streaming」とAdobe社の「HTTP Dynamic Streaming」、Microsoft社の「Smooth Streaming」を解説します。なお、いずれも専用の新しいバージョンのクライアント・ソフトウェアが必要です。

このような新しい流れの中で、より高度な仕組みでストリーミングをユーザ・オリエンテッドに提供しようとする試み^{※8}も出てきています。詳細はAppendix 3で紹介します。

■ HTTPプログレッシブ・ダウンロード・ストリーミング

HTTPプログレッシブ・ダウンロード・ストリーミングの最大の魅力はコストです。WWWサーバやCGI/SSIを使用でき、RTSPやRTPなどのような特別なストリーミング環境を必要としません。また、RTSP DESCRIBE/SDP部分をHTTPのGETメソッ

※6 「平成22年版情報通信白書」第2部第4章第1節1.(3)通信系コンテンツ市場の動向

※7 「平成22年版情報通信白書」第2部第4章第1節2.モバイル・コンテンツ産業の市場規模

※8 参考URL (有)ネットワーク・メンター技術情報リンク：

http://www.network-mentor.com/new_technology.html

CQ出版社

ISBN978-4-7898-4089-7

C3055 ¥3200E

CQ出版社

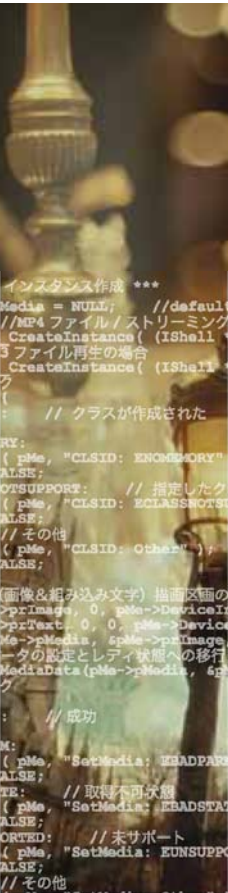
定価：本体3,200円（税別）



マルチメディア・ ストリーミング技術

目次

- 第 1 章 基礎理論
- 第 2 章 マルチメディア伝送基盤の技術標準
- 第 3 章 マルチメディア標準規格
- 第 4 章 マルチメディア・ネットワーク・システムの利用と運用
- 第 5 章 マルチメディア・ネットワーク・システムの実装
- 第 6 章 モバイル・クライアントやストリーミング関連の実装
- 第 7 章 モニタ分析
- 第 8 章 マルチメディア・ネットワークの産業構造と新しい技術の流れ



このPDFは、CQ出版社発売の「マルチメディア・ストリーミング技術」の一部内容見本です。
内容・購入方法などにつきましては以下のホームページをご覧ください。
<http://shop.cqpub.co.jp/hanbai//books/40/40891.htm>