

Title	ユーザジェネレート型コンテンツのためのリアルタイム配信システムの設計とプラットフォームの構築
Sub Title	Design and implementation of real-time broadcasting system for user generated contents
Author	山下, 恭平(Yamashita, Kyohei) 杉浦, 一徳(Sugiura, Kazunori)
Publisher	慶應義塾大学大学院メディアデザイン研究科
Publication year	2021
Jtitle	
JaLC DOI	
Abstract	
Notes	修士学位論文. 2021年度メディアデザイン学 第930号
Genre	Thesis or Dissertation
URL	https://koara.lib.keio.ac.jp/xoonips/modules/xoonips/detail.php?koara_id=KO40001001-00002021-0930

慶應義塾大学学術情報リポジトリ(KOARA)に掲載されているコンテンツの著作権は、それぞれの著作者、学会または出版社/発行者に帰属し、その権利は著作権法によって保護されています。引用にあたっては、著作権法を遵守してご利用ください。

The copyrights of content available on the Keio Associated Repository of Academic resources (KOARA) belong to the respective authors, academic societies, or publishers/issuers, and these rights are protected by the Japanese Copyright Act. When quoting the content, please follow the Japanese copyright act.

修士論文 2021年度

ユーザジェネレート型コンテンツのための
リアルタイム配信システムの設計と
プラットフォームの構築



慶應義塾大学
大学院メディアデザイン研究科

山下 恭平

本論文は慶應義塾大学大学院メディアデザイン研究科に
修士(メディアデザイン学)授与の要件として提出した修士論文である。

山下 恭平

研究指導委員会：

杉浦 一徳 教授 (主指導教員)

砂原 秀樹 教授 (副指導教員)

論文審査委員会：

杉浦 一徳 教授 (主査)

砂原 秀樹 教授 (副査)

南澤 孝太 教授 (副査)

修士論文 2021 年度

ユーザジェネレート型コンテンツのための
リアルタイム配信システムの設計と
プラットフォームの構築

カテゴリ：サイエンス / エンジニアリング

論文要旨

スマートフォンや SNS の普及とともに、コンテンツや流行の流れは目まぐるしく変化している。その中でも、YouTube Live やニコニコ生放送、Twicas といったリアルタイム配信プラットフォームの市場は拡大しており、5G といった広帯域のモバイルネットワークの供用開始により、より高品質やコンテンツの配信も可能となっている。しかし、このようなリアルタイム配信は未だ専門的な知識が必要であり、また高度な処理を長時間行う環境が必要である。本研究では、リアルタイム配信を支援する環境として、BaaS(Broadcast as a Service) やクラウドコンピューティングを用いた、初心者でも扱いやすい配信環境のプラットフォームの構築を行った。本実験ではモバイル端末向けに UI 設計を行い、映像伝送には WebRTC を用いて、5G や 4G 通信下において実験を行った。

キーワード：

コンテンツミキシング, リアルタイム配信, クラウドコンピューティング, 5G

慶應義塾大学大学院メディアデザイン研究科

山下 恭平

Abstract of Master's Thesis of Academic Year 2021

Design and Implementation of Real-Time Broadcasting System for User Generated Contents

Category: Science / Engineering

Summary

With the spread of smartphones and SNS, the flow of content and trends is changing. Among them, the market of real-time broadcasting platforms such as YouTube Live, Nico Nico Live, and Twicas is expanding, and the availability of high-bandwidth mobile networks such as 5G is making it possible to broadcast higher quality content. However, the real-time broadcasting requires specialized literacy and an environment. In this research, we implement a platform for real-time broadcasting that is easy to use even for beginners, using BaaS (Broadcast as a Service) and cloud computing as a support environment for real-time broadcast. The prototype is designed with UI for mobile devices, and experiments are conducted under 4G and 5G communications using WebRTC for video streaming.

Keywords:

Contents Mixing, Realtime broadcasting, Cloud computing, 5G

Keio University Graduate School of Media Design

Kyohei Yamashita

目 次

第1章 序論	1
1.1. ストリーミング	1
1.2. リアルタイム配信の普及と一般化	3
1.3. 問題点	3
1.4. 研究の目的	4
1.5. 本論文の構成	6
第2章 リアルタイム配信	7
2.1. 「リアルタイム型」コンテンツ配信	7
2.2. 配信を行うための環境	8
2.3. 映像伝送技術	11
2.3.1 映像伝送方法	11
2.3.2 コーデック	13
2.4. 配信プラットフォーム	15
2.4.1 従来の配信プラットフォーム	15
2.4.2 インターネットを活用した配信プラットフォーム	16
2.5. コンテンツ	20
2.5.1 ユーザジェネレート型コンテンツ	20
2.5.2 高付加価値コンテンツ	22
2.5.3 オープンでローカリティの高いコンテンツ	23
2.6. ユーザジェネレート型コンテンツにおけるリアルタイム配信	24
2.6.1 視聴者から見たリアルタイム配信	24
2.6.2 配信者から見たリアルタイム配信	25

2.6.3	配信プラットフォームから見たリアルタイム配信	25
2.7.	本章のまとめ	26
第3章	配信の現状および課題	27
3.1.	ライブ配信を行う環境	27
3.1.1	映像コンテンツ制作プロダクションの環境	27
3.1.2	アマチュア制作の環境	28
3.1.3	モバイルデバイスから PC を通じて配信	29
3.1.4	モバイルデバイスから RTMP で配信	30
3.2.	配信に関する知識	32
3.3.	インタラクティブな配信	32
3.4.	従来の方法を用いた実験	34
3.4.1	条件	34
3.4.2	RTSP を用いた方法	34
3.4.3	NDI を用いた方法	38
3.4.4	TVU を用いた方法	39
3.5.	Broadcast as a Service	41
3.5.1	BaaS の事例	42
3.6.	本章のまとめ	45
第4章	クラウドコンピューティングによる配信環境の実現	46
4.1.	MILC(Mixer for Internet Live Contents)	46
4.2.	要件	47
4.2.1	要件1: モバイルデバイス単体での高度な配信の実現	47
4.2.2	要件2: 高品質な配信の実現	48
4.3.	設計	48
4.3.1	クライアントアプリケーション	48
4.3.2	ネットワーク	49
4.3.3	利用方法	51
4.4.	実装	52

4.4.1	開発方法	52
4.4.2	インターフェース	53
4.4.3	構成	54
4.4.4	プログラム	56
4.4.5	ネットワーク構成	56
4.4.6	映像伝送	59
4.4.7	レンダリングコンテンツ	59
4.5.	本章のまとめ	60
第5章	実験及び評価	62
5.1.	概要	62
5.1.1	評価	62
5.2.	KMD Forum 実験	63
5.2.1	実験内容	63
5.2.2	実験結果	64
5.3.	秋葉原実験	65
5.3.1	実験内容	65
5.3.2	実験結果	66
5.4.	定性評価	67
5.5.	定量評価	68
5.6.	本章のまとめ	69
第6章	結論	71
6.1.	本研究において実現した事項	71
6.1.1	技術的要件における解決	71
6.1.2	心理的障壁の解決	72
6.2.	本研究の貢献	72
6.3.	今後の課題	73
6.3.1	配信環境における課題	73
6.3.2	技術的要件における課題	73

謝辭	75
参考文献	76

目次

1.1	YouTube のおすすめ動画紹介ページ	2
1.2	研究テーマ 全体図	5
2.1	「リアルタイム型」配信	8
2.2	プレナリーミーティングでの配線	9
2.3	配信機材の例：ATEM Mini Pro	10
2.4	配信ソフトウェアの例：Open Broadcast Software	10
2.5	ミキサーによるオーバーレイ	11
2.6	コミックマーケット	16
2.7	コミックマーケットにおけるサークルの例	17
2.8	YouTube にて動画を投稿する YouTuber	19
2.9	YouTube チャンネル：例として KMD チャンネル	19
2.10	コンテンツツリー	21
2.11	ゲーム配信の一例	22
2.12	YouTube における投げ銭	23
2.13	オープン性とローカリティ	24
3.1	テレビ局の環境のモデル	27
3.2	TVU を使用した環境のモデル	28
3.3	アマチュア配信の配信環境のモデル	29
3.4	外から配信を行う環境のモデル	30
3.5	ツイキャス配信の画面	31
3.6	Airmix Solo	31
3.7	出力設定	33

3.8	通信速度および電波強度	35
3.9	RTSP 実験 構成	36
3.10	RTSP/NDI 実験を行ったエリア	37
3.11	NDI 実験 構成	38
3.12	TVU 通信構成	39
3.13	TVU 実験を行ったエリア	40
3.14	BaaS モデル	41
3.15	Singular.Live “Composer”	42
3.16	StreamYard スタジオ	44
4.1	MILC モデル	47
4.2	MILC モデル	49
4.3	ネットワークモデル	50
4.4	配信の準備のモデル	51
4.5	配信中のモデル	52
4.6	プログラムモデル	53
4.7	PC 版 MILC	53
4.8	タブレット版 MILC	54
4.9	スマートフォン版 MILC	55
4.10	ソース配置の操作	55
4.11	MILC 構成	56
4.12	ネットワーク構成	58
4.13	レンダリングコンテンツ：地図	60
4.14	WebRTC を用いた通信のモデル	61
5.1	展示構成	63
5.2	展示光景	65
5.3	配信画面	66
5.4	秋葉原実験 実験範囲	68
5.5	秋葉原調査	69

5.6	転送量と解像度の推移	70
-----	----------------------	----

表 目 次

3.1	RTSP 実験の条件	37
3.2	NDI 実験の条件	39
3.3	実験の条件	40
4.1	ホストサーバ (sharo1) スペック	58
4.2	アプリケーションサーバ (sharo3) スペック	59
4.3	ミキシングサーバ (sharo5) スペック	59
5.1	秋葉原実験の条件	64
5.2	秋葉原調査の条件	67

第 1 章

序

論

1.1. ストリーミング

データを送受信する手法として「ストリーム」というものが存在する。ストリームとは物の流れを指す言葉であり、ストリームで映像を受信するというのは、データを連続的に受信し、そのデータを出力装置で表示することを指す。通常のダウンロード再生との違いは、ダウンロード再生はデータを全て受信し、そのデータを記憶装置に記憶した後、その記憶したデータをもとに再生する。そのため、ダウンロードされたネットワークの無い場所でも再生することができ、半永久的に再生することが可能である。それに対し、ストリーム再生は全て受信し終わってなくても受信されたところまで再生することが可能であり、データを記憶装置に記憶する必要がないため、コンテンツ再生のサイクルを早めることが可能である。

YouTube [1] やニコニコ動画 [2] など、動画をストリーム方式により提供するポータルサイトの登場により、好きな動画をすぐに再生することができるようになったため、より動画といった大容量なコンテンツを身近に楽しむことができるようになった。ストリーミングには、Skype [3] やSkYHoleTV [4] などエンドユーザ同士が、PC等のデバイスに専用のアプリケーションをインストールし、お互いに通信を行う「P2P型」コンテンツ配信。配信者がポータルサイトへコンテンツをアップロードし、ポータルサイトのサーバがエンドユーザの要求に応じてコンテンツをストリーミング配信する「サーバ型」の配信、そしてYouTube Live [5] やニコニコ生放送 [6] など配信者がポータルサイトへリアルタイムに制作された映像や音声などのコンテンツをサーバを通じて多数のエンドユーザへ配信を行う「リアルタイム型」の配信がある。動画配信サイトによく見られるのはサーバ型

で、コンテンツをサーバへアップロードするだけで配信を行うことができる手軽さと、図 1.1 に示すように、同サービスを利用するユーザの趣味や行動の特徴により、常に興味の唆られるコンテンツが自動的に紹介され、視聴者にとって興味のあるコンテンツを連鎖的に視聴することができる。また同時に、コンテンツへアクセスする機会が増えるとともに、コンテンツの多様化も進んだ。従来のラジオやテレビといったストリーミング配信と比べ、インターネットを通じた配信はコストが安く、より多様で視聴者のターゲットが絞られた(ローカリティの高い)コンテンツを配信しやすくなった。また、高性能で安価なコンピュータが大衆化し、ソフトウェアの種類が増えたことにより、一般ユーザでもコンテンツを制作するコストや難易度が下がった。特筆すべき特徴として、一般ユーザが自身の趣味や志により制作する「同人」コンテンツがデジタル化し、コンテンツ配信ポータルサイトや SNS(Social Networking Service) を通じて同人ネットワークをインターネット上に形成されたことで、よりオープンでローカリティの高いコンテンツが連鎖的に増加したことが挙げられる。

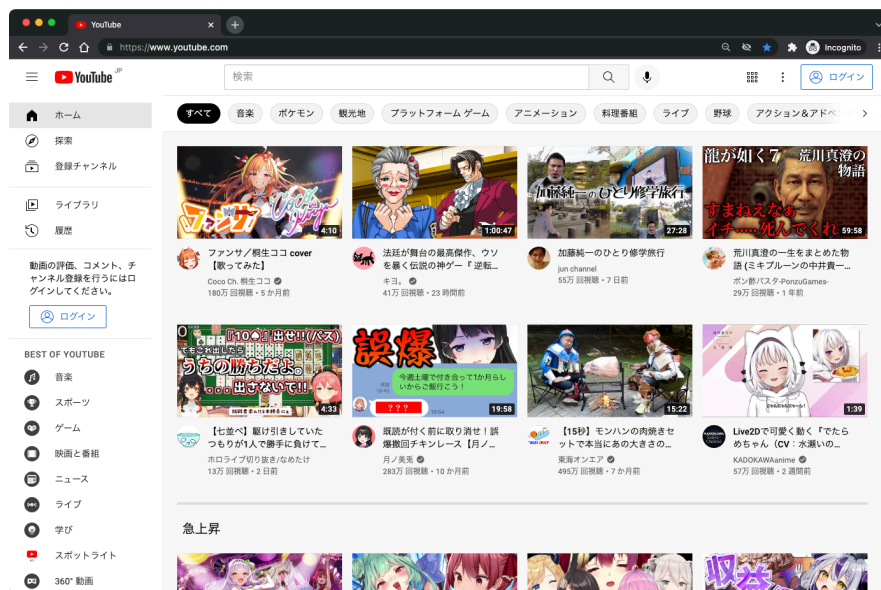


図 1.1 YouTube のおすすめ動画紹介ページ

1.2. リアルタイム配信の普及と一般化

ストリーミング配信方法の一つであるリアルタイム型のストリーミング配信は、画像や動画といったコンテンツやカメラの映像といった映像をリアルタイムにミキシング、エンコーディングを行い、完成形の映像をインターネットや地上デジタル等の電波、中継衛星等を通じてユーザへマルチキャストで配信する配信形態の一つである。その中でも近年ではインターネットを通じた配信が急激に増加しており、スマートフォンなどの個人が常にネットワークへ接続されたデバイスを持ち歩く機会が増え、また4G/LTE/5Gの高速で大容量な移動通信システムの発展により、ユーザが通信容量や速度を気にすることなく、高度なコンテンツを場所を問わず楽しむ可能となった。また、ライブ配信に関する調査 [7] によると、商業系のテレビ放送のみならず一般ユーザによる配信も急激に増加しており、若年層では72.8%がライブ配信を視聴しており、25.7%がライブ配信をしたことがある。特に、TikTokやInstagram Liveといった、より手軽に配信することができるサービスが登場しており、個人の生活や旅行、ゲームの映像など、より個人的な配信が増加しており、その配信内にてコメント機能や電子ギフト機能を用いて視聴者が配信者に対してコミュニケーションを行う準双方向なブロードキャスト配信も多く見られる。例えば、ニコニコ生放送では、配信者のことを「生主」と呼び、視聴者である「リスナー」はコメント機能を使用して生主とコミュニケーションを取る。「凸待ち配信」と呼ばれる配信方式では、Skype [3] 等の通話アプリケーションを用いて生主とリスナーが通話を行い、その他のリスナーがそれに対してリアクションを取るというニコニコ生放送の文化もある。

1.3. 問題点

現在、一般ユーザによる配信の方法には2種類あり、1) コンピュータを用いる方法と2) スマートフォンやWebカメラの映像を直接配信する方法がある。コンピュータを用いる方法では、ミキシングを行うソフトウェアにコンテンツを入力し、コンピュータ上でミキシング、エンコーディングを行う方法である。この方法ではコンピュータ上のソフトウェアを用いて映像のレンダリングやミキシング

を自由に行うことができ、外部のアプリケーションやセンサー等のデータと組み合わせると、インタラクティブな配信を行うことが可能である。しかし、ミキシングやエンコーディングは非常に重い処理であり、高度な配信には高性能なコンピュータと配信に関する知識が必要となる。

デバイスのカメラの映像を直接配信する方法では、煩雑な設定なく配信を行うことができるが、フィルターやエフェクトといった簡易的な処理の設定のみであり、何かと組み合わせる高度な配信を行うことは難しい。この方法は一般的にスマートフォンなどモバイルデバイスで見られるが、画面のサイズが小さいモバイルデバイスでは、同時に何かのコミュニケーションを取ることができず、視聴者に寄り添った配信をすることは難しい。性能においてもモバイルデバイスをPCと同様の方法でミキシング・配信を実現することは難しく、現状では各サービスに最適化されたアプリケーションが必要である。そのためアプリケーションによって出来る表現が異なり、またスマートフォンといったモバイルデバイスから配信することに対して厳しい制約を課しているサービスも存在する。例えばYouTube Liveではチャンネル登録者数が1,000人以上ないと公式アプリケーションからの配信を行うことは制限されている [8]。こういった背景から配信サービスによっては独自の進化を遂げたサイトもあり、配信するコンテンツのジャンルや配信方法に応じてサービスを選ぶのと同時に、サービスの制約がユーザのコンテンツ表現を制限し、また制約の違いから配信サービス間の繋がりが断たれる事態となっている。

1.4. 研究の目的

研究テーマ

図 1.2 に示すように、配信に必要な環境、設備、コンテンツ等をネットワーク上に展開することで、インターネットが扱える場所において自由にアクセスできるようにする。これにより、映像の入力を行うデバイスや操作するコントローラのみで配信を行うことができる他、設備等を用意する必要がなくなることで、より効率的に配信環境を整えることが可能となる。そのためには、配信環境をネッ

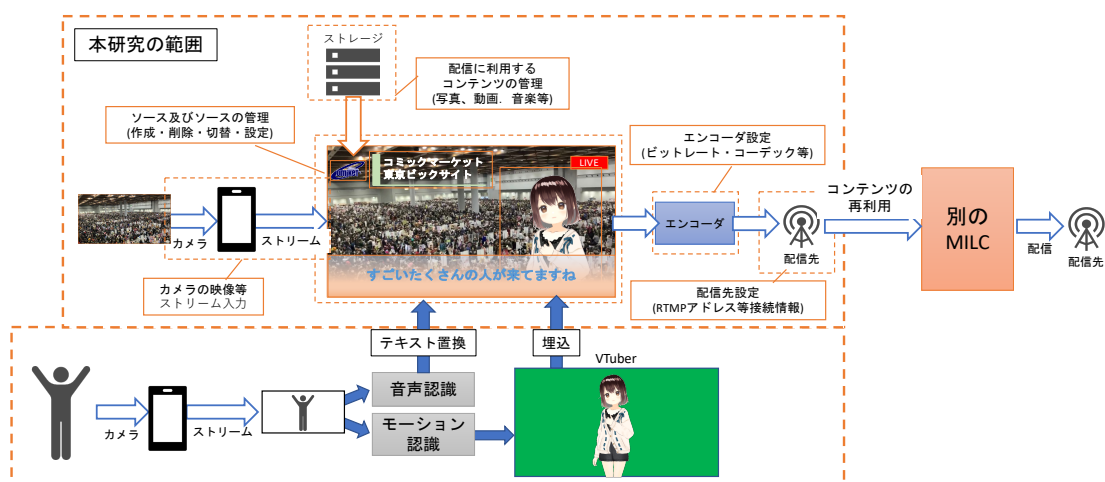


図 1.2 研究テーマ 全体図

トワーク上で扱えるようにし、設定等の項目や入出力を全てインターネットを通して操作できるようにする必要がある、また、機器によるインターフェースや通信方式の違いを解決するため、統一された方法でお互いの機器を操作できるようにする必要がある。

本研究の範囲

本研究では 1.3 節にて提起した問題を解決するため、一般ユーザによるリアルタイム配信において必要な環境をネットワーク上に展開し、クラウドコンピューティングにより提供する。そのため、図 1.2 に示した本研究の範囲の機器を、インターネットにて操作できるようにする。また、前節の問題であった環境構築をサーバ上で行い、デバイス特有の問題や配信に対して要求されるリテラシーを減らすことで、配信の多様性や手法の幅を広げていく。

上記の目的を達成するため、下記の点を考慮して研究を進める。

- 現状の配信に必要な環境や求められるリテラシーの多さを解決させ、ユーザ自身がより自由に表現できる環境を構築する。
- 配信することに対する心理的障壁を取り除く

また、本研究では以下のことを行う。

- 従来の方法を利用した場合の問題点と課題を提起
- モバイルデバイスからの高度な配信を実現するプラットフォームの構築及びプロトタイプの実装
- プロトタイプを用いて実際に配信し評価する

1.5. 本論文の構成

本論文は全6章から構成される。2章で配信の概要を述べ現在の手法をモデルで図示し、3章で既存の配信方法の分析と問題点を述べ、4章で問題を解決する配信システムを定義、実装し、5章で評価実験とその評価を述べ、6章で結論と今後の展望を述べる。

第 2 章

リアルタイム配信

2.1. 「リアルタイム型」コンテンツ配信

リアルタイム配信において、配信者が視聴者に対して直接配信を行う「P2P 型」の配信も存在するが、視聴者の人数が増えるにつれ配信者側の必要な帯域は単純増で増えるため、非常に高いネットワーク帯域が求められる。近年増加しているインターネットにおけるリアルタイム配信は「リアルタイム型」が採用されており、図 2.1 に示すように、配信サーバから視聴者に対してストリームを複製、配信を行うため、配信者側のネットワーク負荷を抑えることができ、配信に専念することができる。また、大半のサービスが配信中のストリームを自動的にストレージへ保存するアーカイブ機能があり、配信中の放送を巻き戻して視聴や、既に終了した配信のアーカイブを「サーバ型」配信のように動画として視聴することができる。

「リアルタイム型」コンテンツ配信を構成する要素は以下が挙げられる。

- 配信を行うための環境
- 配信プラットフォーム
- コンテンツ

それぞれの要素について詳細を表す。

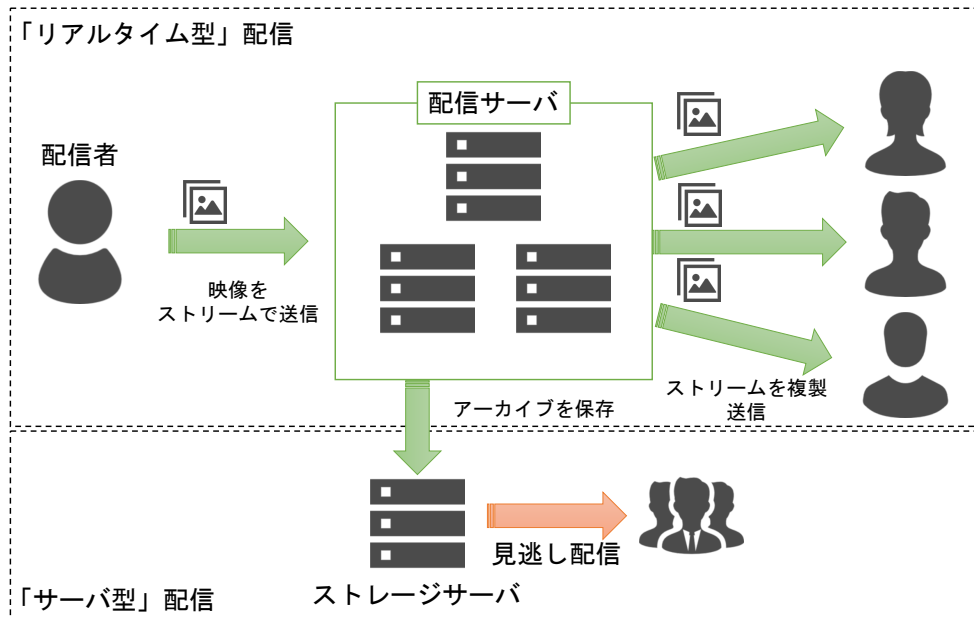


図 2.1 「リアルタイム型」配信

2.2. 配信を行うための環境

配信を行うための環境は様々な機材で構成されている。映像を切り替えるスイッチャー、映像や音声を合成するミキサー、別のインターフェースへ変換するコンバーターなど多種多様である。例として、図 2.2 は慶應義塾大学で行われたプレナリーミーティングの配線である。映像や音声を適正にミキシングし、また発表者のマイクや PC の音量に合わせて調整、発表開始や終了に合わせて映像の切替を行うなど、様々な機器によって支えられている。

機材にはハードウェア上で行うものとソフトウェア上で行うものがある。ハードウェアで行う機器は図 2.3 に示すように、機械に物理ボタンやインターフェースがついており、操作は機器上のボタンをする場合と PC やコントロールする機材と接続し、他機器から操作するものがある。ハードウェア上で処理を行うため、安定した処理が行える他、直感的な配線を行うことが出来る。ソフトウェア上で行う配信環境は、図 2.4 に示すように、コンピュータ上で動作するソフトウェアに、インターネットソースや USB 等のインターフェースで接続した映像や音声を

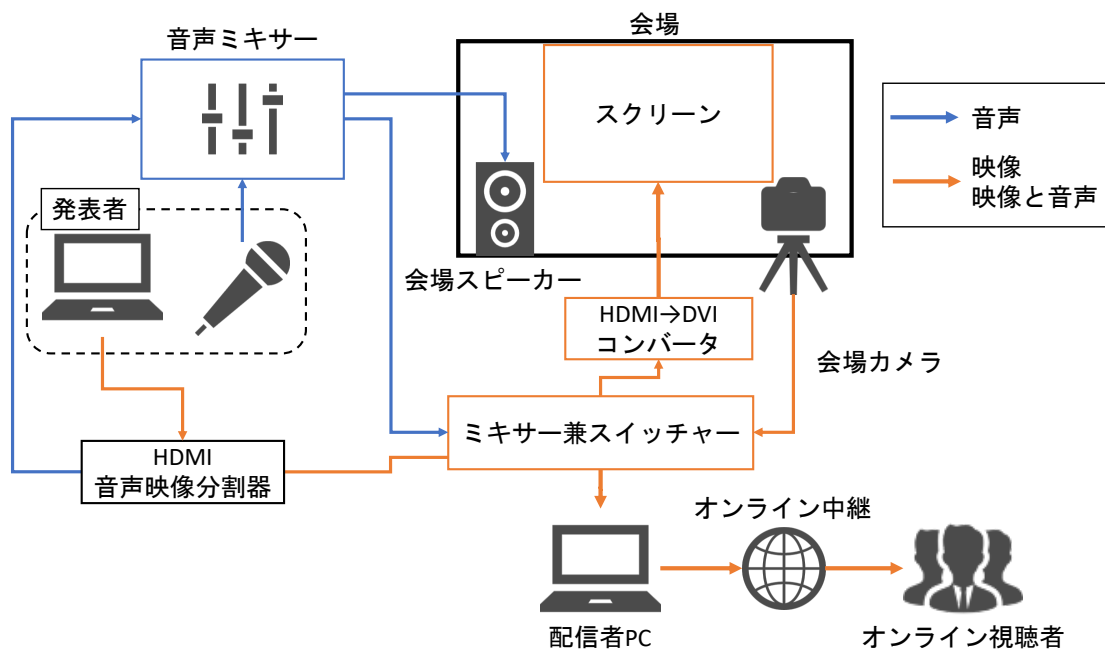


図 2.2 プレナリーミーティングでの配線

入力し、ソフトウェア上で処理、出力を行うものである。ソフトウェア上で完結するためPCがあれば簡単に導入することが出来る。PC上の処理性能を利用し動作するため、複数のソフトウェアとリソースを共同利用される。また、OBSのようなミキサーは、図2.5に示すように複数のコンテンツを順番に重ねて合成することで、映像を作り上げる。ハードウェアの機材からソフトウェアへ入出力を行う場合、コンバータやキャプチャボードと呼ばれる機材が必要となる。これはインターフェースを変換し、お互いの機材で利用することが出来る形へ変換するものである。



図 2.3 配信機材の例：ATEM Mini Pro

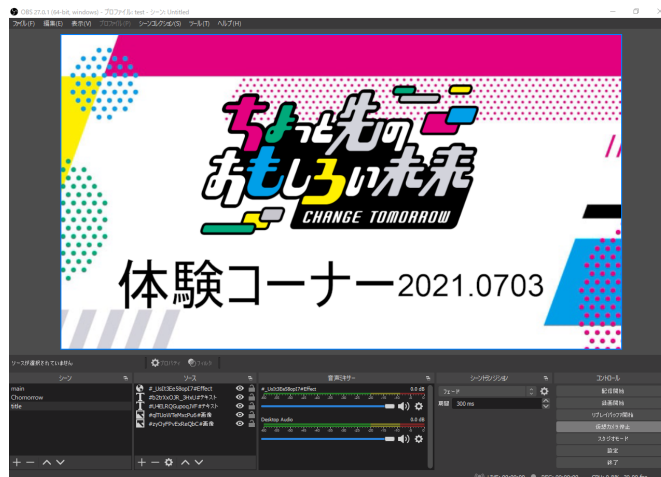


図 2.4 配信ソフトウェアの例：Open Broadcast Software

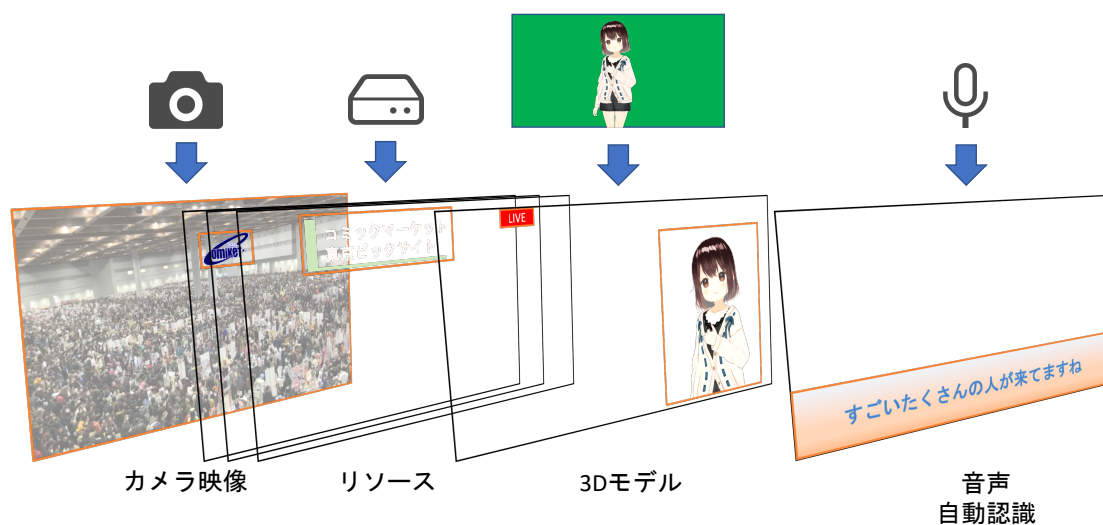


図 2.5 ミキサーによるオーバーレイ

2.3. 映像伝送技術

映像伝送は、伝送する方法として物理ケーブル、あるいはインターネットを通じる方法ではプロトコルが存在する。また映像や音声を圧縮せずに伝送すると容量が大きくなり、転送帯域が足りなくなることから、データを特定の手段を用いて符号化、データの圧縮を行うエンコードを行う。その際に利用される符号化手段がコーデックと呼ばれる。

2.3.1 映像伝送方法

映像伝送には様々な規格が存在しており、使用される状況に応じて最適な規格が変わる。映像伝送する物理的なケーブルでは HDMI や SDI が存在しており、専用ケーブルであることから信頼性が高く、安定した伝送をすることが可能である。ネットワークを利用した映像伝送では、RTSP や NDI, WebRTC などが存在しており、Ethernet あるいはワイヤレスにより伝送することが可能であることから、既存のネットワーク環境を使用して映像伝送を行うことができる。それぞれの規格の詳細を述べる。

HDMI

High-Definition Multimedia Interface(HDMI) は映像や音声を 1 本のケーブルにて送信する通信規格の一種であり, 主にデジタル家電向けのインターフェース規格である. 最新の規格である HDMI2.1 では最大 48Gbps の通信速度があり, DSC1.2 の圧縮技術をサポートしており, 最大で 10K(45760p)/60Hz にて伝送することが出来る.

SDI

Serial Digital Interface(SDI) はビデオ信号伝送規格の一種であり, 高速シリアルインターフェース規格である. SDI 規格は物理層の規格であり, 全米映画テレビジョン技術者協会 (SMPTE) によって策定されている. 現在主流の規格は HD-SDI は SMPTE292M やプログレッシブ HDTV 向けの 3G-SDI は SMPTE424M, 4K 伝送向けの 12G-SDI は SMPTE ST 2082 [9] である. 無圧縮の映像と音声を 1 本の同軸ケーブルにてほぼ無遅延で伝送することができることから, 主に業務用のビデオ機材で利用されており, 最大長が 100m であるため, 映像伝送ケーブルとしては長距離な通信を行うことができる. 最大通信速度は 12Gbit/s であり, 4K 映像を無圧縮にて送ることができる. また, 映像伝送を IP 化した規格もあり, SMPTE 2022-6 や SMPTE 2110 がそれに当たる. トランスポートには UDP を採用しており, 非圧縮または JPEG XS¹により圧縮される. HD(1080p) 時のデータレートは 1.1Gbps 以上あり, 10Gbit 以上の通信環境が必要となる.

RTSP

Real Time Streaming Protocol(RTSP) はネットワーク経由で映像・音声をリアルタイムに配信するためのプロトコルであり, RFC2326 [10] として標準化, 最新の RTSP2.0 は RFC7826 [11] として標準化されている. 基本的にサーバからクライアントへ一方方向に伝送され, メディアサーバからメディアを取り出すような形

1 SMPTE 2110-22

で通信が行われる。データ転送には通常は Real-time Transport Protocol(RTP) が使用される。RTSP は映像伝送の制御を行うプロトコルであり、RTP は映像転送を行うプロトコルに当たる。

NDI

Network Device Interface(SDI) [12] はネットワーク経由で映像・音声をリアルタイムに配信するためのプロトコルであり、HDMI や SDI と親和性が高く、映像配信システムの構築に向いている規格である。NewTek 社により発表され、特定のハードウェアを必要とせず、インターネットを通じて利用することが可能である。圧縮は H.264 または H.265 を使用しており、NDI HX2 では HD(1080p) 映像信号を低遅延リアルタイムで 1Mbps~50Mbps 以下に圧縮することができる。

WebRTC

Web Real-Time Communication(WebRTC) は、ウェブブラウザやモバイルアプリケーション上で使用することが出来るリアルタイム通信の手法である。Justin Uberti や Peter Thatcher によって開発されたオープンソースのプロジェクトであり、P2P 通信により直接ウェブブラウザ間で通信を行うことが出来る。シグナリングには JSEP [13] が使用され、SDP により映像や音声のコーデック情報の交換、データ通信の転送レートを交換する。データ転送の手法ではあるが、映像や音声をストリーム送信することが可能であり、Web ブラウザ上で動作する映像伝送の手法としては現在最も使われている。

2.3.2 コーデック

コーデックはデータを符号化する方法のことであり、エンコードはコーデックを使用して符号化を行う行為のことである。解像度の縦を x [px]、解像度の横を y [px]、チャンネル数を c 、色深度を d 、1 秒間のフレーム数を f 、映像の長さを l [s] と置くと、映像を無圧縮で転送する場合のデータ容量は以下の計算式で表される。

$$S = whcdfI$$

例えば、解像度はHD(1080p)で30frames/s, RGBの3チャンネルで8bitカラーの場合、60分の動画の場合は625.71GB²となり、1.39Gbpsの帯域がないと映像伝送を行うことができない。また実際には音声データも含まれるため、これより大きくなる。

エンコードは映像や音声データを効率良く符号化するため、データが圧縮され効率よく伝送を行うために必要な処理である。符号化を行うエンコードと復号化を行うデコードでは、同じコーデックであっても処理に必要なリソースは異なり、使用するデバイスや映像の種類によって適切なコーデックは異なる。

MPEG-4 AVC

MPEG-4 AVCはH.264とも呼ばれる、Moving Picture Experts Group(MPEG)と言われるISO/IECのワーキンググループによって策定された映像圧縮規格である。ISO/IECではISOではISO/IEC 14496-10:2020 [14]として規定されており、現在の最もメジャーな映像圧縮規格の一つである。動き補償、フレーム間予測、DCT(離散コサイン変換)、エントロピー符号化などを組み合わせたアルゴリズムを利用しており、従来のMPEG-2³と比較して2倍近い圧縮を行うことができる。

HEVC

High Efficiency Video Coding(HEVC)とはH.265とも呼ばれ、H.264の後続として開発された動画圧縮規格の一つである。ISO/IECではISO/IEC 23008 [16]として策定されており、ブロックサイズの適正化など圧縮効率が優れており、H.264と比較して約2倍程度圧縮効率が上がっている。

2 8bit=1byte, 1024byte=1Kbyte, 1024Kbyte=1Mbyte, 1024Mbyte=1Gbyteと計算

3 H.222/H.262, ISO/IEC 13818 [15]

VP9

VP9 は Google によって開発された動画圧縮コーデックの一つである。コンテンツとしては WebM が採用されており、The WebM Project にてドキュメントが公開されている [17]。ブラウザにおいて H.264 に変わるコーデックとしてよく使用されており、WebRTC においては現在積極的に使用されているコーデックの一つである。YouTube においては 4K 動画に対し VP9 及び AV1 にて動画のエンコーディングを行っている。

AV1

AOMedia Video 1(AV1) は AV9 の後続として開発され、非営利団体の Alliance for Open Media(AOMedia) により開発された。現在は標準化されていないものの、GitHub にて標準化に備えたドキュメントを公開している [18]。Web 上において映像に携わる多くの企業⁴が AOMedia に参加しており、ロイヤリティフリーで使用することができる。圧縮方法に RSNR を使用した場合、H.265 と比較して 30% ほど圧縮に優れており、WebRTC にも採用された [19]。

2.4. 配信プラットフォーム

2.4.1 従来の配信プラットフォーム

インターネットの普及以前は本や雑誌、CD や USB などの記録媒体にコンテンツを載せ、お店や展示会を通して行われてきた。デジタルコンテンツの配信では電波を通じてやラジオやテレビなど受信機へ入力し、音や画面に出力する方法が採られている。ユーザコンテンツを配布する場としては即売会⁴が挙げられる。例としてコミックマーケットと呼ばれる同人誌即売会では、図 2.6 に示すような会場に図 2.7 に示すようなサークルと呼ばれるスペースが並べられ、そのサークル

⁴ Cisco, Google, Intel, Microsoft, Mozilla, Netflix, ARM, NVIDIA, Hulu, Facebook, Apple(順不同)(2021/12/14 現在)

に個人やグループで制作した本やアクセサリ，ゲーム等を置いて展示，及び販売する．これらの特徴は実際のスペース内でコンテンツの製作者あるいはサークル関係者が直接販売するため，同じ志を持つ購入者がサークルと直接コミュニケーションを取ることが出来る点である．フィードバックを得られる環境は非常に重要であり，動機づけに関する研究では，ポジティブな感想は物質的な報酬より内発的動機を高めると言われている(桜井，1984) [20]．



図 2.6 コミックマーケット

2.4.2 インターネットを活用した配信プラットフォーム

インターネットが普及してからは掲示板や SNS(Social Networking Service) を通じてコンテンツが公開されることが多くなっている．インターネットを通じて公開されるデジタルコンテンツは多岐に渡り，イラストや動画，音楽やゲームといったプログラム，3D モデルなど様々なジャンルが存在する．このような完成形のデータを配布するプラットフォームはいわゆる「サーバ型」配信で行われることが多く，対してライブカメラの映像やリアルタイムに制作されたコンテンツをストリームとして配信するプラットフォームは「リアルタイム型」である．「リアルタイム型」配信のプラットフォームは多数存在しており，配信する番組のジャ



図 2.7 コミックマーケットにおけるサークルの例

ジャンルや配信者の種類に応じて様々な特徴を持つ。現在あるプラットフォームの例は以下の通りである。

- 商業系
 - Abema TV [21]
 - DAZN [22]
 - 17LIVE [23]
- 一般ユーザ向け
 - Instagram ライブ [24]
 - ツイキャス [25]
 - TikTok [26]
- 商業系と一般ユーザ向けが混在
 - YouTube Live
 - ニコニコ生放送
 - LINE LIVE [27]

これらの配信サービスの特徴として、配信者や配信するコンテンツのジャンル等に応じてチャンネル(またはマイページ)を持ち、そのチャンネルの中で番組を配信する。例えばYouTubeでは、図2.8に示すように、YouTuberと呼ばれる視聴者の興味関心を示すコンテンツを制作している人が多数おり、図2.9のような自身が持つ(または所属する)チャンネルにおいて、コンテンツを投稿している。これらのチャンネルは前2.4.1小節にて例をあげたコミックマーケットでのサークルと同様の意味合いを持ち、ユーザの興味関心を得られるコンテンツや、趣味や経験を共有するコンテンツを並べるスペースがオンライン上に容易されており、そこに並べる形式である。またコンテンツ視聴ページでは視聴数・コメント・高評価数が表示され、コンテンツに対するフィードバックを投稿する機能があり、視聴数やチャンネル登録者数(フォロワー数)といった値はクリエイタの評価として重要な意味を持つ。

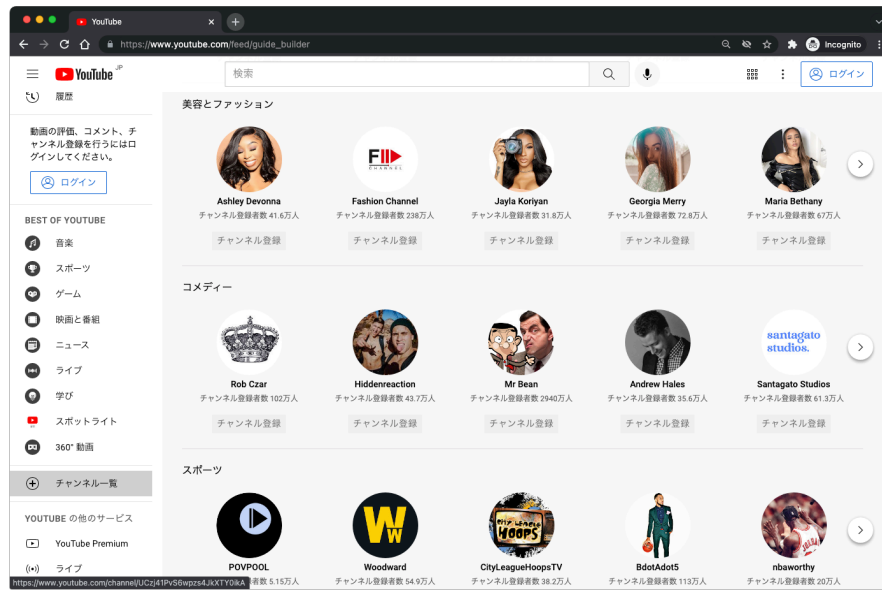


図 2.8 YouTube にて動画を投稿する YouTuber

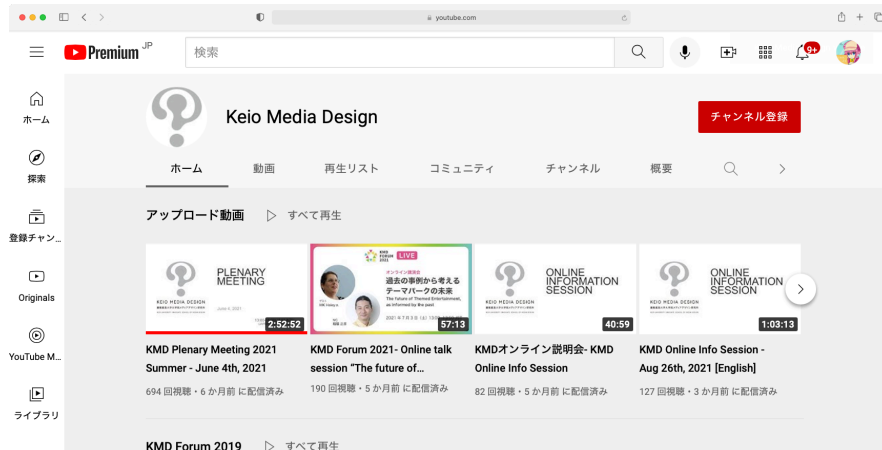


図 2.9 YouTube チャンネル：例として KMD チャンネル

2.5. コンテンツ

2.5.1 ユーザジェネレート型コンテンツ

ユーザジェネレート型コンテンツ (User Generated Contents)(UGC) は User Created Contents(UCC) や Consumer Generated Media(CGM) と呼ばれることもある。プロダクション会社により制作されたコンテンツではなく、個人の手によって作られたコンテンツである。これらのコンテンツには特定の出版社などパブリッシャーは存在しておらず、不特定なプラットフォームにて投稿されていくものである。ブログやSNSによる投稿、その中でのコメントやレビューもユーザジェネレート型コンテンツと言える。インターネットが普及する以前では、オックスフォード英語辞典の未登録用語の補完のための読者による公的貢献が有名である。インターネットが普及すると、気軽に情報を送信することができるようになり、またそれを自動的に処理して記録するサイトが現れたことにより、ユーザジェネレート型コンテンツは急速に増加し始める。基礎プラットフォームはニュースグループといった、ジャンルや目的別によって分けられた掲示板によるユーザの書き込みにより、ユーザによるコミュニティやコンテンツネットワークの形成である。いわゆる同人作品も、同じ趣味や志を持つ者によるコミュニティやネットワークが形成され、その中で様々な方向性を持つ人によって作品(コンテンツ)が作られている。例えばコンテンツの利用範囲を明確にし、ユーザ間のコンテンツ利用を促進させているニコニ・コモンズ [28] では、図 2.10⁵に示すように、より良いコンテンツにしていくため、誰かの作ったコンテンツをもとにコンテンツが作られ、それをまた別の人利用するように、連鎖的に様々な人がコンテンツを作り続けられている。ユーザコンテンツを利用して作成されたユーザコンテンツの例として、図 2.11 は筆者が行ったゲーム配信の画面である。

5 <https://commons.nicovideo.jp/material/nc237092> のスクリーンショットである

コンテンツツリー：ウチもやったんだからさ

コンテンツツリーとは、今見ている作品を作るために利用した作品（親作品）や、逆に今見ている作品を利用して作られている作品（子作品）などをツリー的に表示する機能です。

今見ている作品を作るために利用した作品

今見ている作品

今見ている作品を利用して作られている作品

他のユーザーが今見ている作品を親作品として登録する事で、追加されていきます。

親作品 (3)

子作品 (2)

1-2件 前へ 次へ

図 2.10 コンテンツツリー



図 2.11 ゲーム配信の一例

2.5.2 高付加価値コンテンツ

配信されるコンテンツには、配信するコンテンツに価値をつけて配信を行うコンテンツの有料配信が行われている。商業として企業が有償にてコンテンツを配信する形態に限らず、前項の一般ユーザによる有料配信及びドネーションタイプの配信もあり。配信コンテンツに対して高付加価値をつける方法も多数ある。商業として企業が行うコンテンツ配信には大きく3つに分かれる。AVOD(Advertising Video On-Demand)では視聴する際に広告を表示し、その広告料によりコンテンツ制作及び配信されているものである。主にテレビの地上デジタル配信やYouTubeがこの形式を取っている。SVOD(Subscription Video On-Demand)では定額支払うことにより、ポータルサイト内のコンテンツへのアクセスが自由に行える形式である。Netflix [29]やdアニメ [30]などがある、PPV(Pay Per View)は都度課金にてコンテンツを提供する形態であり、視聴するごとに課金が発生する。レンタルDVDショップや映画館で採られている従来のコンテンツ提供方式である。

特にその中でも特筆すべき機能として、ライブ配信サービス(投げ銭等)の動向整理によると [31],「投げ銭」機能と呼ばれる視聴者が配信者に対して電子的なアイテムや金銭を送る機能がある。図2.12に示すように、YouTubeでは自動翻訳

機能により、リアルタイム配信であっても世界中の人が視聴，そして世界中の通貨単位で投げ銭が行われる。

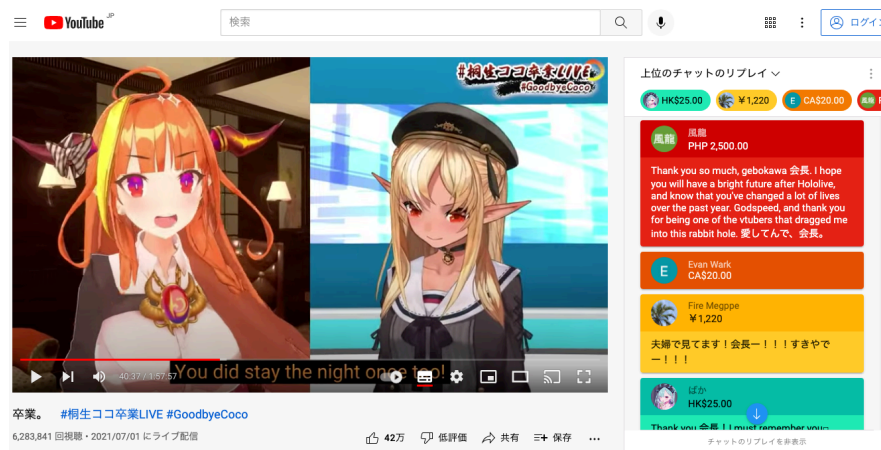


図 2.12 YouTube における投げ銭

2.5.3 オープンでローカリティの高いコンテンツ

リアルタイム配信によって配信されるコンテンツは一度に複数のユーザへ同一内容が配信される。コンテンツの中にはアクセスの制限があるコンテンツや、特定の人に向けたコンテンツも存在する。図 2.13⁶はデータのオープン性を横軸にローカリティを縦軸に、コンテンツの種類によってプロットしたものである。例えば地上デジタル放送の場合、テレビ塔から周辺のアンテナに向け電波を一斉送信するため、データのオープン性は非常に高く、また番組の視聴率をあげるため誰もが理解できるコンテンツであるため、ローカリティは低いと言える。UGCのような個人が同じ趣味を持つ人に向けて作られ SNS 等にて公開されるコンテンツは、誰もがアクセスできるためオープン性が高く、また共感を求めて作られるコンテンツなため、ローカリティは非常に高いと言える。対して前 2.5.2 小節で述べた高付加価値コンテンツは、UGC のローカリティはそのままオープン性を一転させ、対価を支払った特定の個人に限定される。この様なピンポイントな内容の

6 自室分散型コンテンツ情報管理アーキテクチャ（三島，2012）[32] を元に作成

コンテンツが幅広く存在するのがUGCの特徴であり、配信においてもローカリティの高い配信は、他の配信と差別化するためにも非常に重要となる。

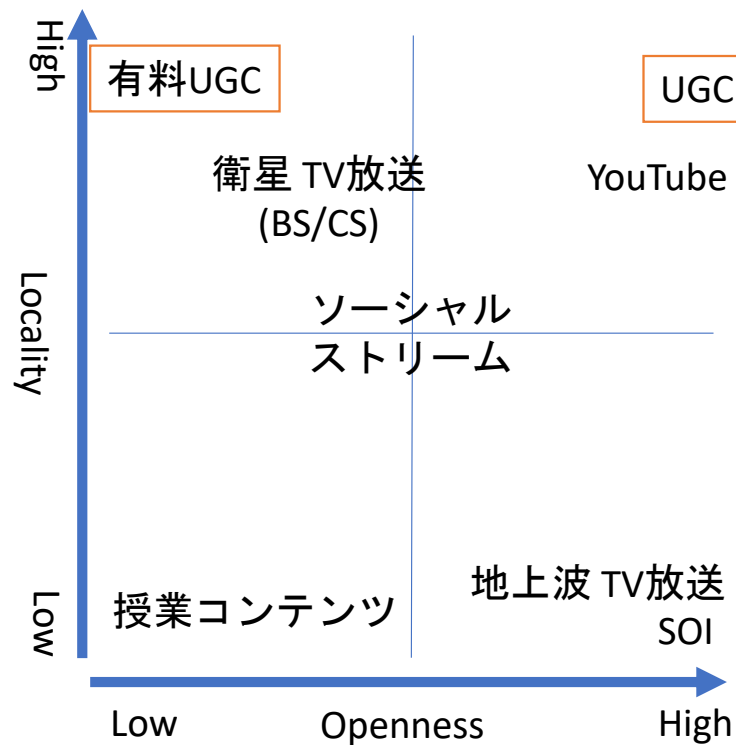


図 2.13 オープン性とローカリティ

2.6. ユーザジェネレート型コンテンツにおけるリアルタイム配信

2.6.1 視聴者から見たリアルタイム配信

視聴者が求めるコンテンツは興味や好奇心の唆られるコンテンツである。プラットフォーム上には大量のコンテンツが投稿されるが、その中で視聴者が言語的問題がなく、理解することが出来る難易度、求めているコンテンツのジャンルが一

致し、なおかつ視聴に至るコンテンツは限りなく少ない。一般的にプラットフォームによりメディア形態や方向性が異なるため、ユーザの方向性に合致したプラットフォームにアカウントを持ち、その中で活動する時間が大きく占める。お気に入りの配信者を見つけた際はチャンネルをフォローし、その配信者による配信が始まった際にはリピーターとして視聴する。コメントやリクエスト機能を使用し、配信者とコミュニケーションを取りつつ、良いコンテンツに対してはその対価(スポンサー)として金銭を支払うこともある。

2.6.2 配信者から見たリアルタイム配信

配信者は自身の趣味や考えをアウトプットする手段としてリアルタイム配信を行う。その際により多い視聴者を獲得するため、視聴者のフィードバックをもとに求められた要求に答える必要がある。また、配信の視聴者はリピーターの割合が多く、リピーター率を増やすことはチャンネル登録者数を増やすことが視聴者を増やす重要な鍵となる。また、多数あるチャンネルから自身のコンテンツを視聴してもらう為には、他と差別化されたコンテンツであり、なおかつ付加価値の高いコンテンツである必要がある。こうした経緯より、高品質な配信を行うためには戦略が重要であり、より高度な配信を行うためにはより高度な機材や知識などの環境を整える必要がある。

2.6.3 配信プラットフォームから見たリアルタイム配信

配信プラットフォームは多数の配信が集まり、配信者と視聴者をマッチングさせる場である。アクティブユーザを増やすには、配信の細かい情報が必要であり、タイトルや説明文、タグや配信ジャンルなど様々なプロパティをもとに、視聴者の動向に合わせた配信を常に紹介し続ける必要がある。また、配信を楽しむ環境を整えるのも配信プラットフォームの役割である。よりインタラクティブな配信が行えるよう、コメントやアイテムなどのコミュニケーションツールを用意し、配信者及び視聴者が飽きない仕組みを整える必要がある。インプレッションの多い配信者はプラットフォームから見ると重要な資産である。17LIVEでは配信者に

対して独占的な配信を約束させる契約を行っており、対価として有利な報酬率や給料を与えている。

2.7. 本章のまとめ

本章では、「リアルタイム型」コンテンツ配信を構成する要素を「配信を行うための環境」「配信プラットフォーム」「コンテンツ」の3要素に分解し、それぞれの実例と現在の手法を述べ、「リアルタイム型」コンテンツ配信の構造と複雑化を述べた。それぞれの要素が時代とともに変化していき、機能が強化されると共にやらないといけない設定や要求が複雑化している。特にユーザジェネレート型コンテンツ (UGC) はローカリティの高い内容であることが特徴であり、あらゆるジャンルが幅広く存在している。こうしたコンテンツを配信するにあたり、配信者と視聴者がお互いに強く結びつき、インタラクティブな配信にしていくことが重要である。

第 3 章

配信の現状および課題

3.1. ライブ配信を行う環境

3.1.1 映像コンテンツ制作プロダクションの環境

テレビ局などの映像コンテンツ制作プロダクションの場合，スタジオに一通りの機材が揃っており，映像を撮影するカメラマンとシーンの切り替えや映像や音声の調整を行う映像編集者がいる．出演者含め，それぞれの役割ごとにその役割に専念することができ，スムーズに配信を行うことが重要視される．映像制作を行う環境も重要であり，映像や音声途切れることは許されないため，いかなる状況でも一定以上の品質が保証される衛星中継が今でも主流である．その一例を図 3.1 に示す．

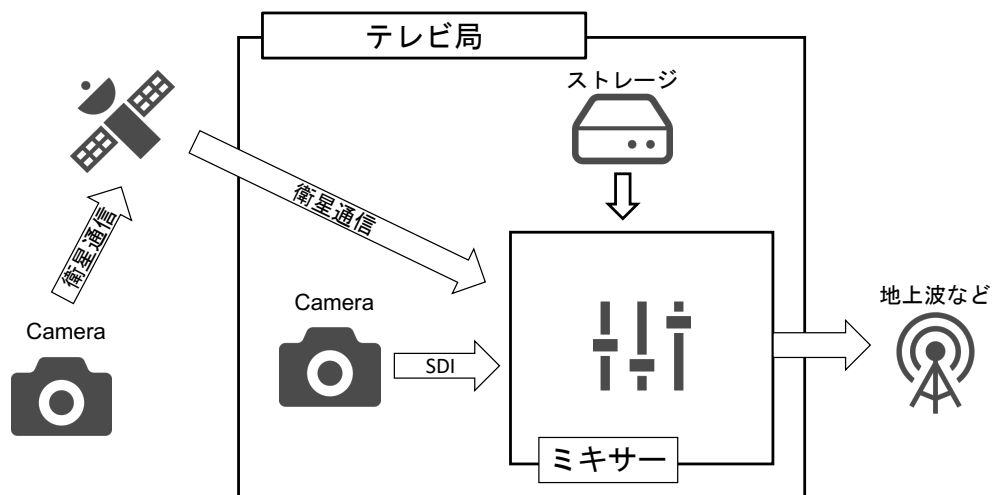


図 3.1 テレビ局の環境のモデル

また、4G などのキャリア回線を利用した技術も登場しており、TVU [33] や LiveU [34] がある。例として、日本における TVU では、日本の大手キャリア回線を複数接続し、信号の強さをもとにダイナミックにパケットをルーティングさせることで通信の安定化と品質の担保を行っている [35]。通信経路の例を図 3.2 に示す。

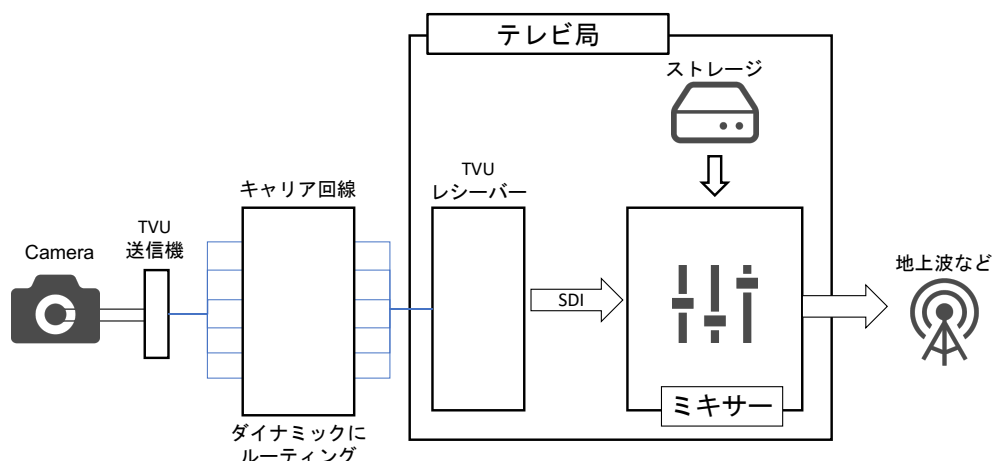


図 3.2 TVU を使用した環境のモデル

3.1.2 アマチュア制作の環境

アマチュア配信では基本的に一人でも配信が可能な環境が大半であり、現在主流な方法ではコンピュータにカメラやマイクを接続し、コンピュータ上のソフトウェアでミキシング、エンコーディングを行い、配信を行う方法である。配信に必要な環境の一例を図 3.3 に示す。一般的に接続するカメラやマイクなどのデバイスはローカル環境にあるものが大半であり、接続方法は USB や AUX、場合によっては Ethernet であることが大半である。

ミキシングを行うソフトウェアには Open Broadcast Software(OBS) [36] や Split-medialabs Ltd. 社が公開する XSplit [37] などがある。このソフトウェアを使用し、映像や文字などのソースを合成、H.264 といったコーデックでエンコードを行う必要があるが、これらの処理は CPU/GPU リソースを大量に消費するため、安

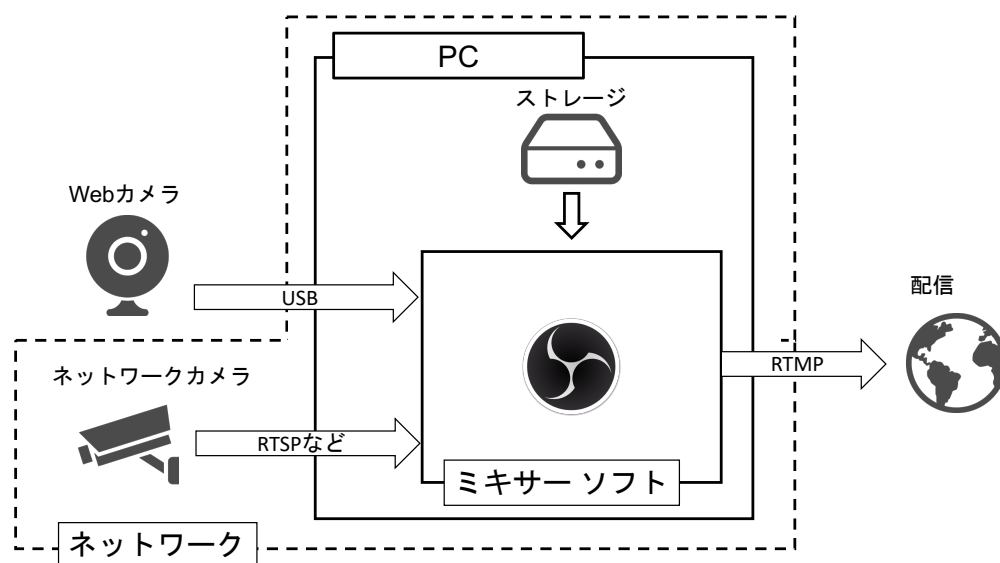


図 3.3 アマチュア配信の配信環境のモデル

定して配信を行い場合，相当の性能を有したコンピュータが必要である．ミキシング等をせず，そのままカメラの映像を配信するだけの場合，ブラウザのみで配信することが可能なサービスもある．

近年ではスマートフォンといったモバイルデバイスから配信するケースも増えている．アプリケーションからスマートフォンのマイクとカメラの映像に簡単な加工をして配信するものが大半であり，複雑な配信を行うことは厳しい．また，外から高品質で複雑な配信を行う場合，ノートパソコン等を外へ持ち出し，直接キャリア回線へ接続，またはテザリング経由で配信を行うことになる．そのためパソコンやその他機材を配信中は常に携帯する必要がある．その際の環境の例を図3.4に示す．

3.1.3 モバイルデバイスからPCを通じて配信

また，スマートフォンなどモバイルデバイスから配信を行うこともできるが，画面サイズの限られたデバイスでは複雑な設定を行うことは難しい．図3.5はツイキャスの配信アプリケーションの配信中の画面である．この画面は配信光景，コ

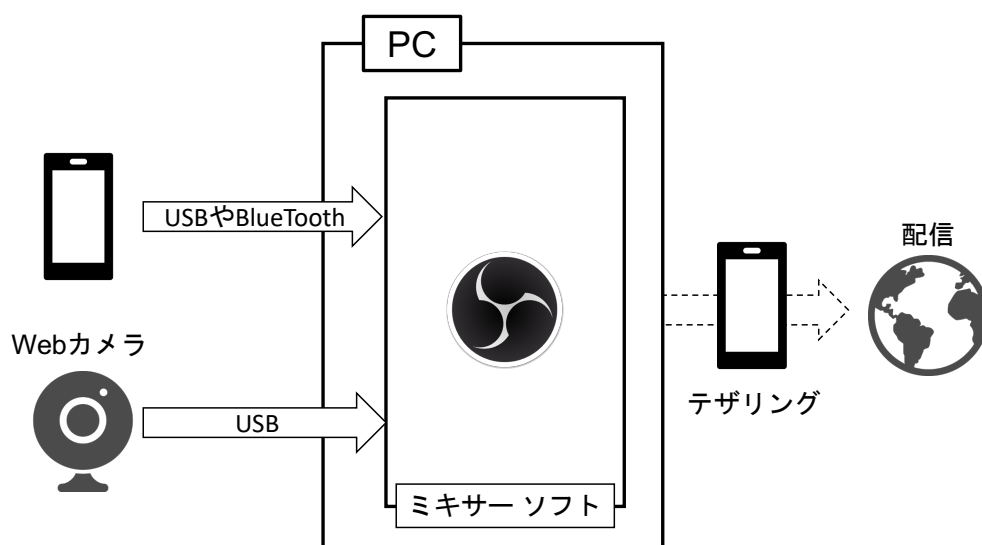


図 3.4 外から配信を行う環境のモデル

メントビューアの2つで構成されている。限られた画面サイズに大量の項目を一度に表示することは難しく、確認できる情報は必要最低限である。また、配信設定やカメラ及びマイク設定が可能ではあるが、パソコンでの配信と比べると非常に設定項目が制限されており、基本的にはカメラの映像にフィルターをかけてストリーム送信する機能しかない。

3.1.4 モバイルデバイスから RTMP で配信

スマートフォンから直接 RTMP として映像配信を行うことのできるアプリケーションとして Airmix Solo [50] や Larix Broadcast [51] がある。例として Airmix Solo の場合、映像や音声品質をコントロールすることができ、デバイスや通信環境に応じて調整すれば、各サービスの専用アプリケーションより高品質な映像で配信を行うことが可能である。また簡単なソースの合成を行うことも可能であり、図 3.1.4 に示す様に、文字や画像をオーバーレイにて組み合わせることも可能であった。しかし直感的な操作を行うことは出来ず、ソースのサイズや配置はピクセル単位で指定する必要があるため、映像や音声品質のコントロールもビットレート指定であるため、ネットワークの知識が要される。またミキシング、エンコード、

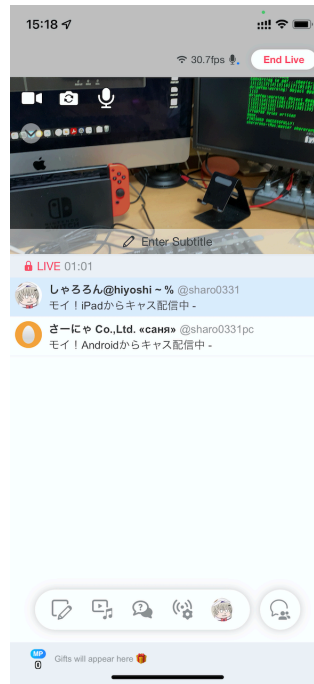


図 3.5 ツイキャス配信の画面



図 3.6 Airmix Solo

ストリーム送信を同時に行うため、熱により動作が不安定となり、長時間の配信を行うことは出来ない。

3.2. 配信に関する知識

配信をスムーズに行うためには、環境構築を行うための知識や、配線やネットワークに関する知識が必要である。例えば、シーンやソースの設定では、Webカメラに対して背景を透過する場合はクロマキーに関する知識、ネットワークカメラの映像を入力する場合はRTSPやNDIのソースを取り込む為の知識、配信をする際にはRTMPの設定や解像度、ビットレートの調整など設定が多数ある。図3.7は一例として、出力の設定画面である。こうした複雑な設定の背景には、入出力する規格が多数存在しており、その規格に合わせて手動で設定しないといけないためである。物理的に配線を行う場合、ケーブルの種類や規格、バージョンによって適合した構成にする必要があり、またデバイスによってインターフェースが異なる場合、コンバーターや変換ケーブルを用いて適合させる必要がある。外部カメラを取り入れる際は、ネットワークに関する知識も必要となる。NDIやRTSPを使用する場合、カメラのIPアドレスを指定する必要があるが、外部ネットワークのソースを取り入れる場合、NAT等を使用して外部からアクセス出来るようにするか、VPNを使用して同一ネットワーク内に見立ててアクセスする必要がある。こうした煩雑な規格や構成を構築するには経験が必要であり、リアルタイム配信を行う心理的な障壁の一つになっている。

3.3. インタラクティブな配信

配信は様々な要素に支えられており、ただ映像を配信するだけではエンターテインメント性の高い配信を行うことはできず、自己満足で終わってしまう。配信された映像は多数のユーザが視聴し、アクティブな視聴者はフィードバックを残す。こうした配信者と視聴者の間には常に相互作用しており、リアルタイム配信においては、リアルタイムにコミュニケーションが取られる。こうした視聴者及び配信者に

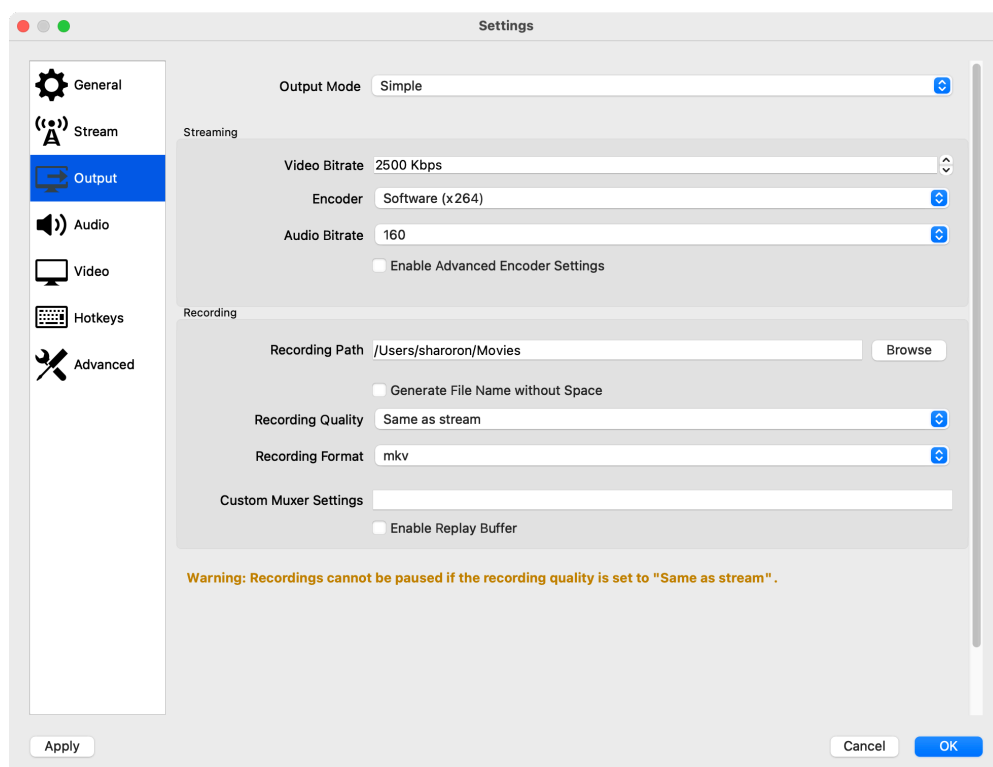


図 3.7 出力設定

とってインタラクティブな配信にするための環境には、自動化とツールの存在が必要不可欠となる。例えば、Voiceroidによる自動読み上げソフトウェアとして「ゆかりネット」[38]や「棒読みちゃん」[39]、視聴者によるコメントを自動処理するソフトウェアとして「マルチコメントビューア」[40]や「NicoNamaCommentViewer」[41]がある。

3.4. 従来の方法を用いた実験

本節では従来の手法、プロトコル、設備を用いて配信環境を構築し、配信を行った。現在の配信モデルの問題点を抽出し、検証を行う。

3.4.1 条件

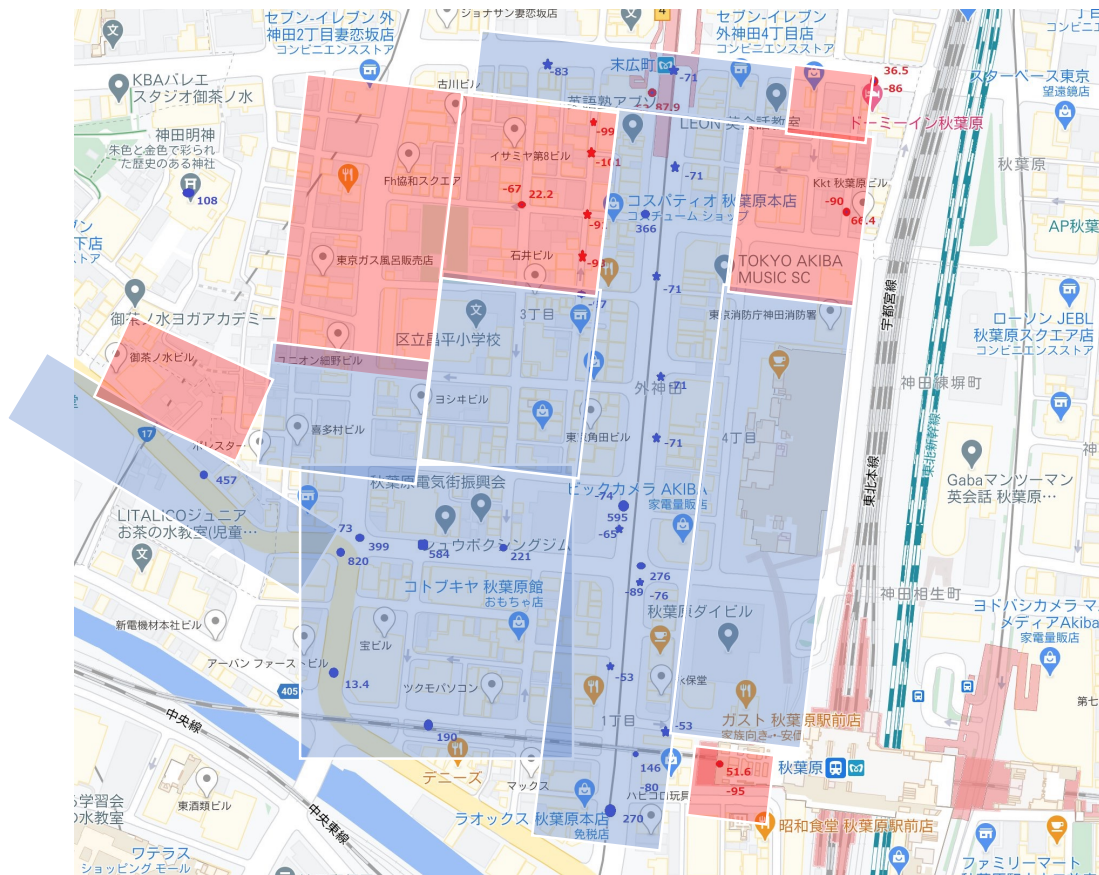
本章の実験では、従来の手法を用いて配信を行い、現状の問題点を抽出する。実験を行った秋葉原エリアの5G対応エリアは700MHz, 1.7GHz, 3.4GHz帯である。電波強度は-100dB~-60dBであり¹、通信速度は100Mbps~800Mbpsであった²。そのため、キャリア回線の通信品質が本実験におけるボトルネックになったとは考えづらい。実際に2020年11月13日13時頃に計測した通信速度及び電波強度は図3.8に示す。

3.4.2 RTSPを用いた方法

映像をストリーム送信する手段としてRTSPはメジャーなプロトコルの一つである。カメラ等の映像ソース側にサーバを立て、映像を要求するクライアント側がサーバに対して映像を要求し、ストリーミングを行うプル型のストリーミングプロトコルとなっている。RTSPはOBSやXSplitなどの一般的なミキシングソ

1 iPhone 12 Proにてキャリア通信のテストモード(電話にて3001#12345#と入力)にし、電波強度を取得した。

2 speedtest(<https://speedtest.net>)による計測。



正の値はDown速度(Mbps)
負の値は電波強度(dB)

5G 青いエリア
4G 赤いエリア

図 3.8 通信速度および電波強度

ソフトウェアでは殆どのケースでサポートされている。当実験では RTSP プロトコルを用いて OBS に入力する際に必要な構成と手間について調査した。RTSP の通信をモバイル回線を介してサーバへ入力するには、モバイルデバイスの IP アドレスを知る必要がある。モバイル回線の IP アドレスは状況や場所に応じて変化するため、今回は VPN サーバを活用することでローカルネットワークに見立てて通信を行った。構成は図 3.9 に示す。

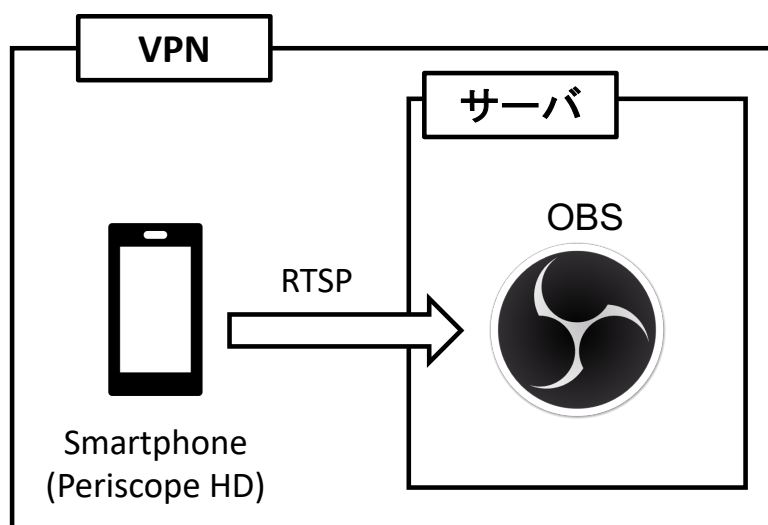


図 3.9 RTSP 実験 構成

また、実験は 2020 年 11 月 14 日に秋葉原エリアの中央通りを 1 周した。実際に歩いた範囲は図 3.10³にて点線で示した。これらの範囲は実験で使用した Softbank 回線の 5G エリア内である [43]。詳細の条件については表 3.1 にて示す。

iOS では RTSP ビュアは多数あるものの、iPhone を RTSP サーバとして動作させるアプリケーションは限られており、VPN を通じて動作させた場合、通信が切断された際に OBS が自動的に再接続されないという問題が発生した。また、5G/4G エリアを跨ぐ場合、ネットワークは一度切断され、ミキシングサーバ側でリロードを行う必要がある。IKEv2 の再接続機能を用いて、RTSP のプロトコル上の通信を常時接続した状態にできないか試みたが、効果はなかった。

3 電子地形図 25000(国土地理院) [42] を加工して作成した。



図 3.10 RTSP/NDI 実験を行ったエリア

表 3.1 RTSP 実験の条件

日時	2020 年 11 月 14 日
デバイス	iPhone 12 Pro
エリア	秋葉原エリア
使用回線	Softbank 4G & 5G
VPN	L2TP+IPSec & IKEv2
アプリケーション	Periscope HD [44]

3.4.3 NDIを用いた方法

NDIとはNetwork Device Interfaceの略で、NewTek社が提唱するビデオ信号伝送規格の一つである[12]。IP通信上で映像を伝送することができるものであり、高品質で映像を伝送することができる。NDIもRTSPと同様にミキサーで扱うことのできるプロトコルの一つであり、OBSへはプラグインを導入することによりサポートされる[45]。また、NDIもRTSPと同様に映像ソース側にサーバを立て、映像を要求することによりサーバからクライアントに映像が送信されるプル型のストリームである。基本的にローカルネットワークで使用することが想定されており、モバイル回線からミキシングサーバへ入力する場合、VPNの接続が必須である。また、NDIはmDNSを用いてクライアントを識別しているため、Discovery Serverという別ネットワークに存在するデバイスを認識するサーバが必要となる[46]。当実験で使用した構成は図3.11に示す。また調査を行った範囲はRTSP実験と同じ図3.10である。

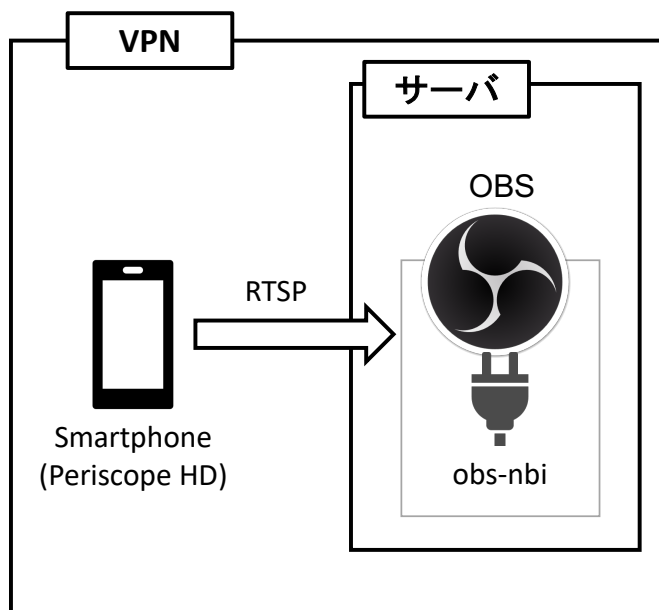


図 3.11 NDI 実験 構成

結果はNDIクライアントを認識することは可能ではあるものの、安定した通信を行うことは難しく、実用性は低いと感じられた。また、Discovery Serverに対応

表 3.2 NDI 実験の条件

日時	2020年11月14日
デバイス	iPhone 12 Pro
エリア	秋葉原エリア
使用回線	Softbank 4G & 5G
VPN	L2TP+IPSec & IKEv2
アプリケーション	NDICam [47]

するクライアントアプリケーションは実験当初では有料アプリケーションのみであり，本研究の手軽な配信には不向きである。

3.4.4 TVU を用いた方法

TVU は docomo や au などのキャリアネットワークを利用した無線映像伝送システムである。サーバ側にレシーバサーバを用意し，カメラ等に送信機を取り付けて映像をレシーバサーバに対して送信するプッシュ型のストリームである。プロフェッショナル向けであり，専用機材のレンタルに1日数万円必要 [48] など費用が高額なため個人では扱いきれないが，比較として実験を行った。本実証ではサーバセンター内にある TVU レシーバに対して映像伝送を行っている。本実験で使用した構成を図 3.12 に示す。実験の条件の詳細を表 3.3 に示す。

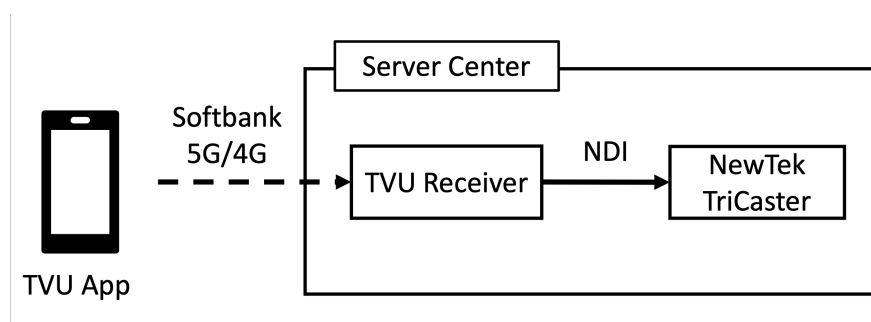


図 3.12 TVU 通信構成

表 3.3 実験の条件

日時	2020年11月12日
デバイス	iPhone 12 Pro
エリア	秋葉原エリア
使用回線	Softbank 4G & 5G
アプリケーション	TVU [49]

本実験も秋葉原エリアにて実験を行った。ルートは図 3.13 に点線で示す⁴。



図 3.13 TVU 実験を行ったエリア

結果は正常に映像を送信することはできたものの、4G/5G エリアを跨ぐ移動を行ったところ数秒の暗転が見られた。また使用したアプリケーションは積極的に HEVC/H.265 エンコードを行い通信量を削減するようになっており、1 時間を超える長時間使用を行ったところ、過負荷によるサーマルスロットリングが発生し、パフォーマンスの低下やブロックノイズの発生、TVU 受信機への接続が不安定になった。

⁴ 電子地形図 25000(国土地理院) [42] を加工して作成した。

3.5. Broadcast as a Service

クラウドコンピューティングによる配信環境の一つとして、Broadcast as a Service(BaaS)という仕組みがある。BaaSとは「サービスとしての配信」であり、配信に必要な機能、例えばネットワーク、エンコード、デコードやストレージといった各機能をオンデマンド方式にて提供するサービス形態である [52]。BaaS のモデルを図 3.5 に示す。ユーザが必要なおきのみ使用することが可能であり (オンデマンド方式)、全ての処理や映像伝送が IP 通信で行われるため、ネットワークに接続されていれば場所を問わず利用することが可能である。また、各機能がエコシステムで動作するため高い拡張性があり、必要な機能だけの利用、または追加に必要な機能があれば組み込むことが用意である。さらに、配信に関する環境をサービスとして提供するため、インターネットを経由してミキサー等の設定が可能である。そのため、カメラは現地で撮影する必要があるものの、編集に関してはオンラインにて行うことが可能である。

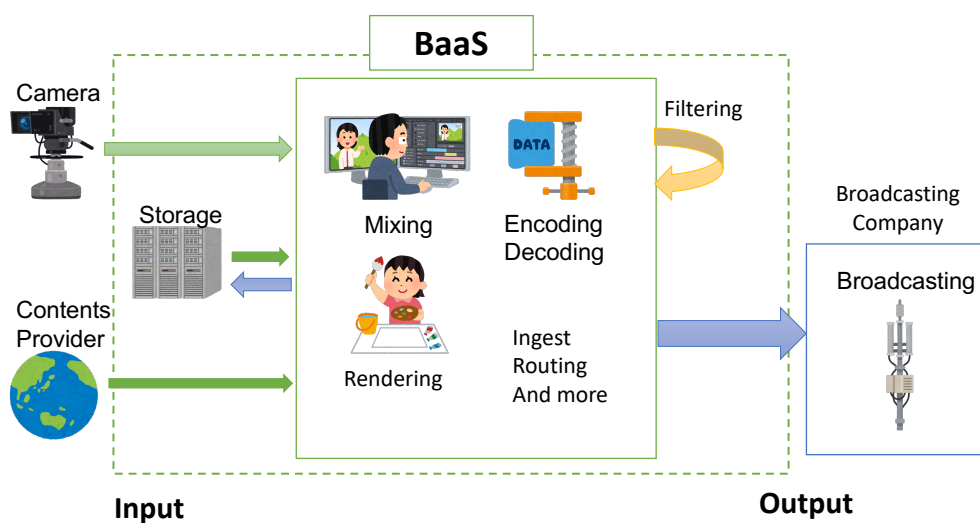


図 3.14 BaaS モデル

3.5.1 BaaS の事例

現在，この BaaS は主に放送業界で活用されている．この仕組みを利用したサービスは以下があげられる．

- Singular.Live [53]
- LTN [54]
- StreamYard [55]

Singular.Live

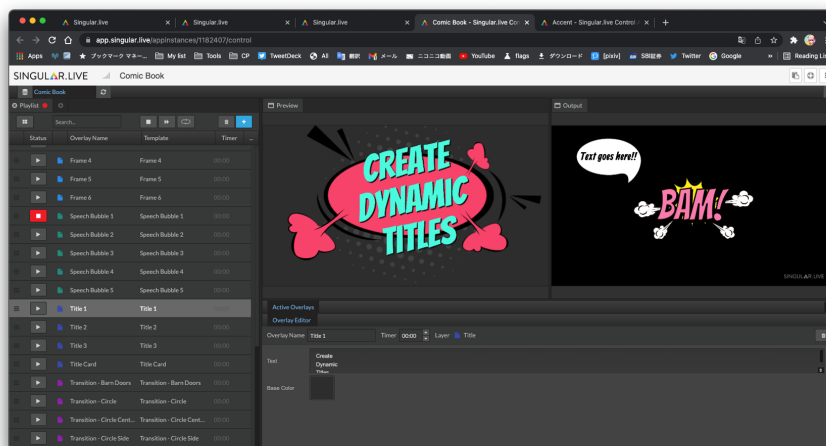


図 3.15 Singular.Live “Composer”

Singular.Live は Singular.Live 社によって提供される，クラウドベースの放送映像制作プラットフォームである．スポーツのライブ中継やニュースの映像に特化されており，オーバーレイ方式のテンプレートを使用することで映像制作を行う．出力は SDI や NDI，ストリーミングに対応している．オーバーレイの編集は図 3.5.1 に示すような “Composer” アプリケーションを通して行う．テンプレートを参考に，テキストや画像等のソースを設置・設定する．配信中は “STUDIO” アプリケーションにて画面の設定・切替，ソースのプロパティ変更等を行う．基本的

にオーバーレイの切替やソースのプロパティを変更することで画面に変化を加えることはできるものの、動的にレンダリングされるコンテンツの表示は行うことはできない。また、映像制作プロダクション向けのツールであり、PCからの利用を想定された作りとなっている。

LTN

LTNはLTN Global社によって提供される、クラウドベースの映像プロダクションプラットフォームである。コンテンツの提供、ルーティング、変換、配信を行うことができる。コンテンツチェーンを利用したワークフローによる一元化された映像制作に力を入れており、クラウド上にてコンテンツをアグリゲートすることにより、効率的なライブコンテンツを含む映像制作を行うことができる。主に制作すべきコンテンツが概ね決まっており、映像制作をワークフローにより自動化することに着目しており、フレキシブルな映像制作を行うことは難しい。また、複雑な映像編集にはプロ用機材と組み合わせて使用することを想定されており、全ての機能をクラウド上で行うことはできない。

StreamYard

StreamYardはStreamYard, Inc.社によって提供される、クラウドベースの映像ミキシングプラットフォームである。図3.5.1に示すようなスタジオで配信の構成を編集する。カメラやスクリーン、プレゼンテーションをソースとして入力し、配信するレイアウトを選択することで画面構成を切り替える。またブランドロゴをオーバーレイにて合成することができる。映像の入力はWebブラウザを通して行うため、遠隔地にいるゲストを参加させやすいという特徴を持つ。レイアウトは決められた中から選択する方式であり、配信者が自由なスタイルで配信を行うことはできず、主にメインとなるコンテンツが決まっており、発表者やコメントータをワイプとして表示するスタイルである。ソースを自由に配置することはできないため、限られた配信スタイルでのみ有用である。

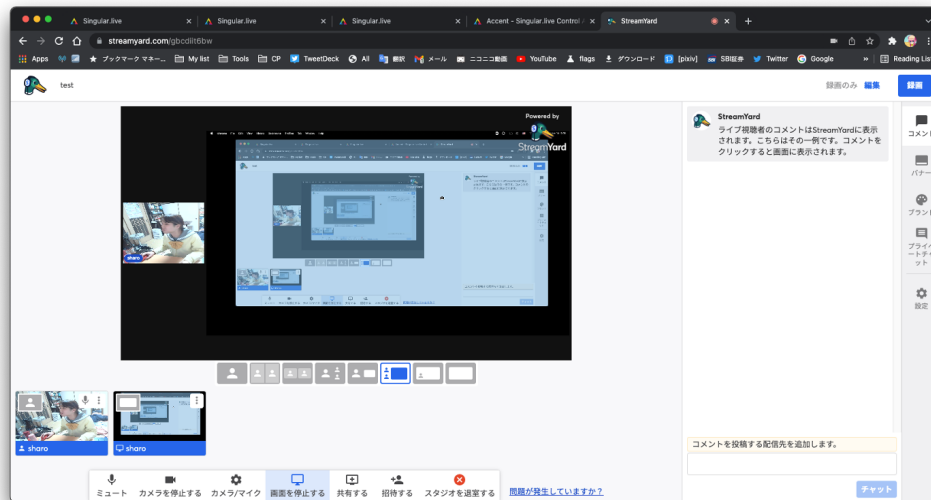


図 3.16 StreamYard スタジオ

これらの特徴

これらはいずれも放送局といったプロフェッショナル用途を想定しており、TVプログラムの制作のための高度なミキシング環境や、外部のコンテンツプロバイダからのデータをスムーズに取り入れられる仕組みとなっている。またプロフェッショナル用途で主に使われる映像伝送プロトコルはNDI(Network Device Interface)やSRT(Secure Reliable Transport)が使われており、テレビ局向けのカメラに専用モジュールを差し込むことで映像伝送を行うことができるようになっている。しかし、これらはプロフェッショナル向けの料金形態をしており、大規模な環境で映像制作を行う場合は良いが、個人が利用する場合は非常にコストの高いものとなっている。また、これらの環境では映像伝送の同期やクオリティの担保を重要しており、冗長性や可用性のために細かい設定を行うことを前提としており、高度なネットワークの知識も必要である。

3.6. 本章のまとめ

本章では、現在使われている「リアルタイム型」コンテンツ配信のモデルを配信者毎に分け、現状と課題を述べた。映像制作プロダクション会社とアマチュア制作では求められている点や目指している点が異なっており、また制作から配信手段まで手法が大きく異なっている。アマチュア制作の環境にて、高度な機器を使用することはコスト面から難しく、限られた環境では配信にて表現できる幅が狭まる。特に屋外から移動通信ネットワークを利用して配信する場合や、モバイルデバイス単体で配信を行う場合、配信プラットフォームやアプリケーション、プロトコル等により、求められるリテラシーが高度になり、また制約等により表現できる幅が狭まる。より良い配信環境にするためには、場所や時間を問わず配信することができ、配信者は配信に専念しつつ、視聴者の求めているコンテンツを常に提供できる統合型の環境が必要である。現在使われているプロトコル(RTSP, NDI, TVU)を使用し、移動通信ネットワークからミキシングし、配信を行った際の現状と問題点を述べた。RTSPやNDIでは、基本的に単一ネットワークから利用することを想定されており、VPNといった仮想的に同一ネットワークから利用していると見せかける必要があった。その際であっても、安定した接続を行うことはできず、移動通信ネットワークの種別が切り替わると通信が切断され、再接続は手動で行う必要があった。TVUを用いた手法では、移動通信ネットワークからであってもスムーズな接続及び配信を行うことは可能ではあったものの、長時間の配信には耐えることが難しい。

第 4 章

クラウドコンピューティングによる 配信環境の実現

4.1. MILC(Mixer for Internet Live Contents)

3章で示した問題点である 1) 配信環境が煩雑なため、環境を構築するために必要な設備の数が多いことや、要求されるリテラシーが多いこと、2) モバイルデバイスからの配信は処理性能が足りず、複雑な配信が不可能であることを解決する為、配信の経験が浅いユーザでも利用することのできる、配信機材や配信に関するリテラシーの要求を減らした環境を構築する。また、総務省によると [56]、一般ユーザの所有しているコンピュータはスマートフォンが最も多いため、スマートフォン単体でも動作する配信環境を整える。従来のローカル環境に配信用の環境を構築するものではなく、クラウドコンピューティングを利用し、配信するための環境をサーバ上に構築し、必要な特に必要に応じてユーザが利用できるオンデマンド形式にて提供する。構造は後述する Broadcast as a Service を参考にした。スマートフォンは性能が PC と比べて乏しく、また高負荷な処理を長時間行うことは難しい。スマートフォン上だけにミキシング環境を整えることは性能や安定性の面で現実的ではないため、ミキシングに必要なパフォーマンスリソースをクラウドコンピューティングにて補う形で環境構築を行い、その新たな配信環境を MILC(Mixer for Internet Live Contents) と名前をつけた。MILC を使用することで、外部コンテンツとの連携も可能にした、インターネットがあれば場所を問わず配信を行うことのできる「リアルタイム配信環境」を実現する。

開発するアプリケーションはクライアント側とサーバ側に分かれ、サーバ側で

はミキシングやエンコーディングを行い、クライアントアプリケーションはサーバへ命令を行うコントローラのような役目を担当する。また、ミキシングに必要なコンテンツの入力等も行う必要がある。

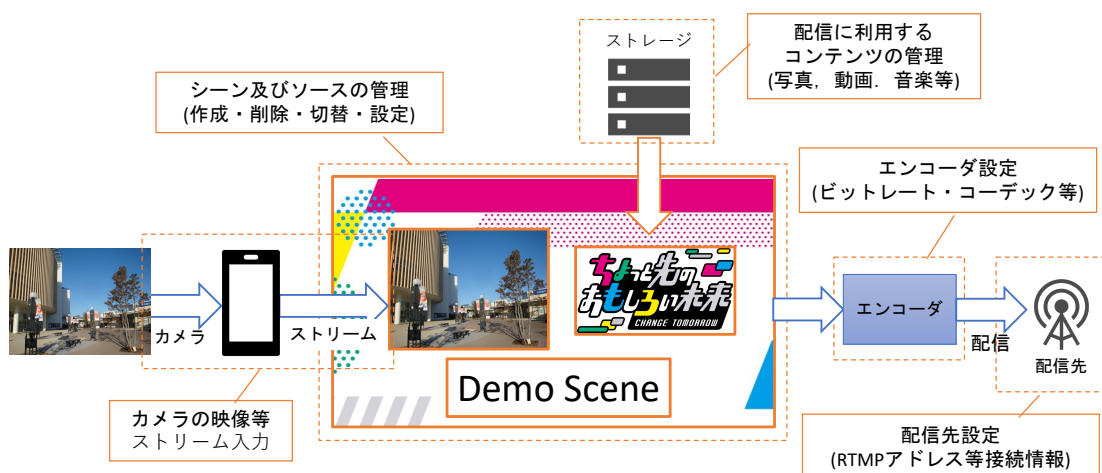


図 4.1 MILC モデル

4.2. 要件

4.1 節にて述べた新たな配信環境を実現するため、以下のような要件を定義する。

4.2.1 要件 1：モバイルデバイス単体での高度な配信の実現

配信に必要な機材を減らし、求められるリテラシーを補助する環境を用意するにあたり、一般ユーザの所有率が最も高いスマートフォンにて PC と同レベルの配信を行える環境を実現する。ミキシングやエンコーディングに必要なコンピュータリソースはサーバ上で行い、スマートフォンは、サーバ上で動作するミキシングソフトウェアを遠隔操作するコントローラとして利用する。また、カメラから入力した映像をミキサーへストリーム送信する仕組みを用意することで、カメラとして扱うことができるようにする。

4.2.2 要件2：高品質な配信の実現

品質の高い配信を安定して行えるようにするため、ミキシング環境をエコシステムにて構築し、ミキシングを行うサーバをユーザ毎に独立した環境にする。ミキシングサーバをユーザ毎に独立して構築した場合、複数ユーザが並列して配信をすることが可能となり、またお互いのミキシング環境間で連携することでコラボラティブな配信を可能とする。また、ミキシングサーバを独立させることにより動的なコンテンツをミキサー上で安全に動かすことができ、よりインタラクティブな配信を可能とする。

4.3. 設計

4.3.1 クライアントアプリケーション

クライアントアプリケーションは以下のような機能を有する。

- ミキシングソフトウェアを操作する
 - － シーンの作成, 切替, 管理
 - － ソースの追加, 削除, 管理
- エンコーダの設定
- 出力先の設定
- コンテンツの管理
 - － 写真や動画, 音楽ファイルの管理
 - － ストリーム入力の管理

基本的には、図 4.2 に示すように、クライアントアプリケーションからミキシングソフトウェアの機能をリモートで操作できるようにすることで、直感的にミキサーをコントロールできるようにする。但し全ての機能をモバイルデバイスか

ら利用することは難しい。動的にレンダリングされるコンテンツに対しては代替手段として、ミキサー上で Web アプリケーションを動作させることで解決する。また、1.4 節にて定義した研究目的である「配信に対する心理的障壁」を取り除くには、煩雑だと思える操作をできる限り減らす必要がある。そのためモバイルデバイスでも直感的な操作ができるよう、ユーザインターフェースや機能の最適化をする必要がある。

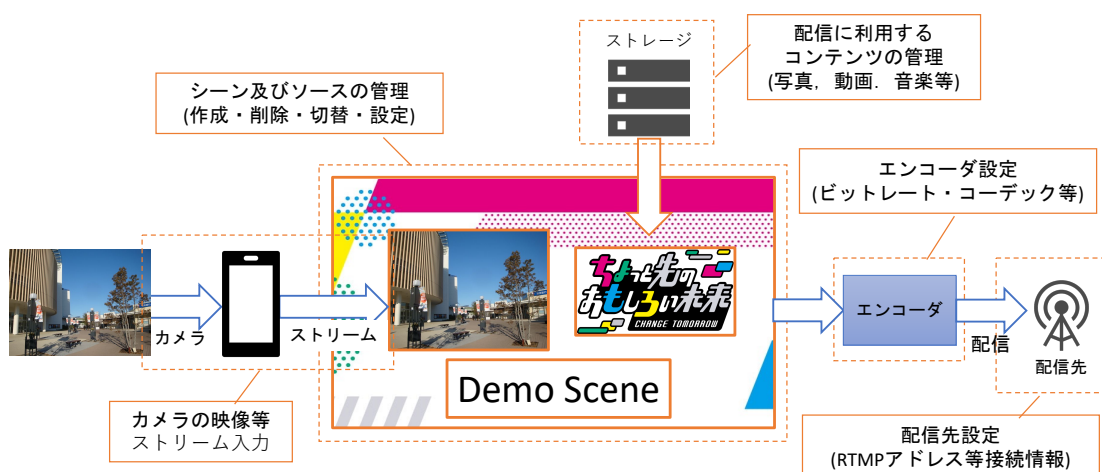


図 4.2 MILC モデル

4.3.2 ネットワーク

BaaS のモデルでは各機能をエコシステムにて動作させる。MILC でも各機能を独立したサーバ・アプリケーションで動作させ、状況に応じた機能の結合を容易にさせる。必要なサーバは以下の通りである。また、サーバ構成のモデルは図 4.3 に示す。複数ユーザが独立してミキシングを行うことができるよう、ミキシングサーバを複数台用意し、配信毎に使用していないミキシングサーバを割り当てる。コントロールサーバはミキシングサーバの割当を管理し、ユーザが適切なサーバへアクセスできるよう支援する。

- Web サーバ

- クライアントアプリケーションを動作させるためのプログラム及びリソースを提供する。
- コントロールサーバ
 - ユーザ毎にアクセス権限を管理，命令や接続するミキシングサーバをコントロールする
 - 配信毎に稼働していないミキシングサーバを割り当て，使用可能にする
 - クライアントがミキシングサーバへ直接アクセスする必要がある場合（カメラ入力など），ミキシングサーバへの接続情報を提供し，通信できるようにする
- ストレージ
 - 配信に必要な動画や画像などのコンテンツを記録する
- ミキシングサーバ
 - ミキシングソフトウェアを稼働させ，配信プラットフォームへ配信を行う

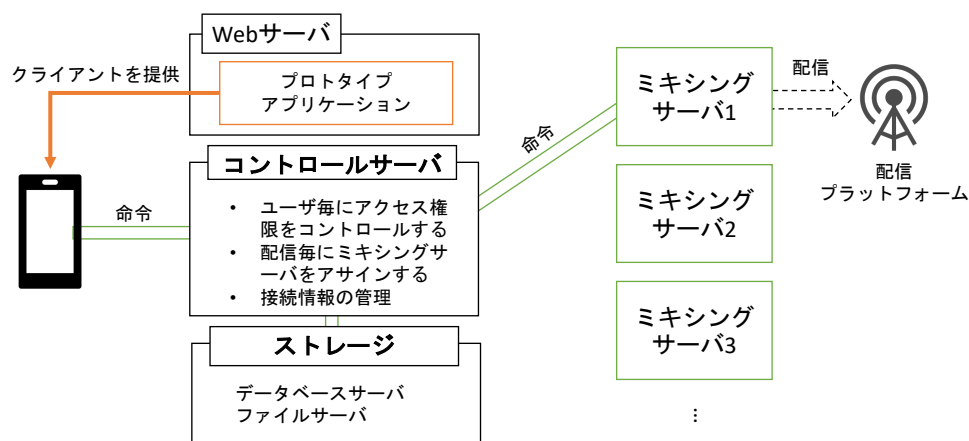


図 4.3 ネットワークモデル

4.3.3 利用方法

配信の手順は準備、配信、終了の順番に行われる。それぞれの手順のモデルをフローチャートにて定義する。まず準備のフローチャートは図4.4に示す。配信を行う前に、配信するプログラムの内容を構成し、それに応じてシーンを事前に準備する必要がある。シーンの追加・設定は配信中でも行うことはできるが、円滑な配信のためには事前に準備をするか、配信中にミキサーを操作する専用の担当者の用意が不可欠である。その次にソースの設定を行う。カメラ映像の場合だと入力元の設定、テキストだとフォントやサイズの設定、写真や動画だとファイルパスの指定等である。状況に応じてソースにフィルターを設定する必要もある。最後に出力先の設定である。YouTube Live やニコニコ動画など、一般的な配信プラットフォームで使用されるプロトコルはRTMPだが、その設定にはアドレス、ストリームキー、アカウント名、パスワードを指定する必要がある。準備が完了したら配信中の操作に移行する。

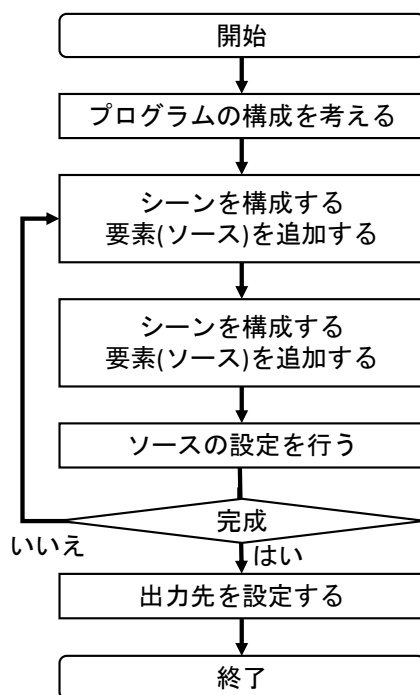


図 4.4 配信の準備のモデル

次に配信中のフローチャートは図 4.5 に示す。配信中は基本的にコンテンツ制作に集中することになるが、必要に応じてミキサーを操作する必要がある。大規模な配信の場合、演者の他にミキサーを操作する担当の人やディレクターがいる場合もある。これらの操作は配信と同時並行で行われることになるため、配信中の操作をできるだけ最適化し手順を減らすことが重要となる。

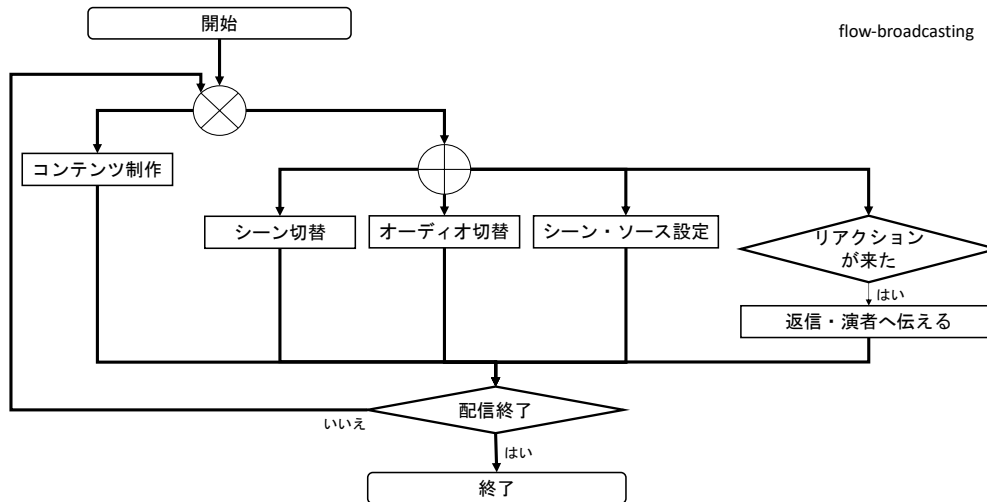


図 4.5 配信中のモデル

4.4. 実装

4.4.1 開発方法

クライアントアプリケーションは様々なプラットフォームで動作するよう Web アプリケーションで開発した。開発言語は TypeScript と JavaScript であり、ライブラリ郡として React や Material-UI を使用した。サーバアプリケーションの開発言語は TypeScript と JavaScript であり、Node.js 環境で稼働する。認証は REST API で行い、その他の命令はリアルタイム性を重要視するため、WebSocket を利用しクライアントとサーバのやり取りを行う。構成のモデルは図 4.6 に示す。

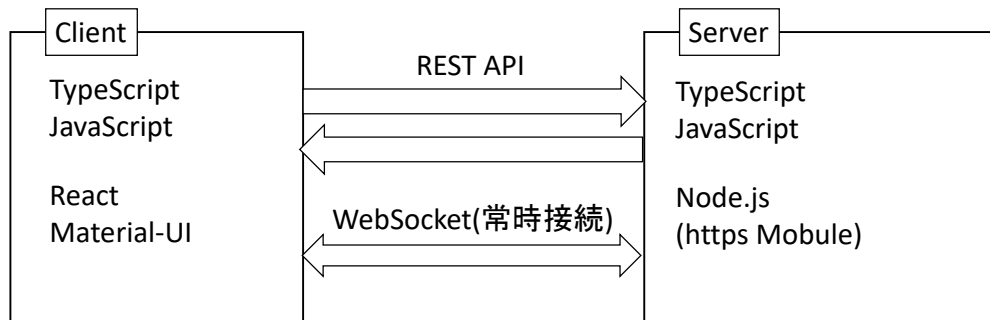


図 4.6 プログラムモデル

4.4.2 インターフェース

様々な環境でを使用することを想定し、レスポンスなデザインでUIを設計した。アプリケーションの横幅に応じてコントロールパネルやメニューの表示を調整している。PC、タブレット及びスマートフォンで利用した際のメインの画面の表示を図4.7、図4.8、図4.9に示した。メインの画面ではシーンの切替、ソースの設定、エフェクトの適応を行うことができるが、横幅の狭いスマートフォンにおいてはタブ表示している。

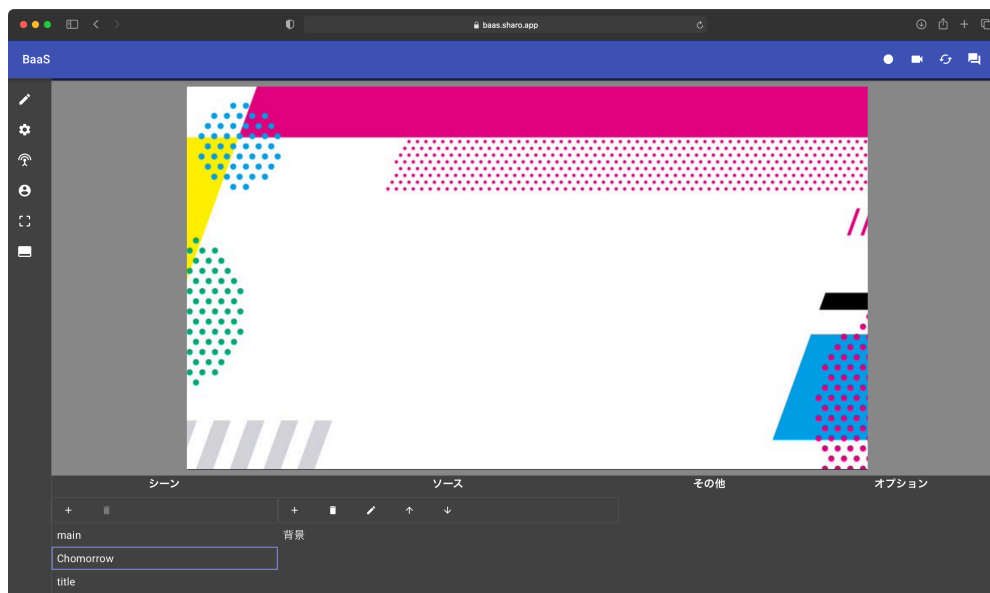


図 4.7 PC版 MILC

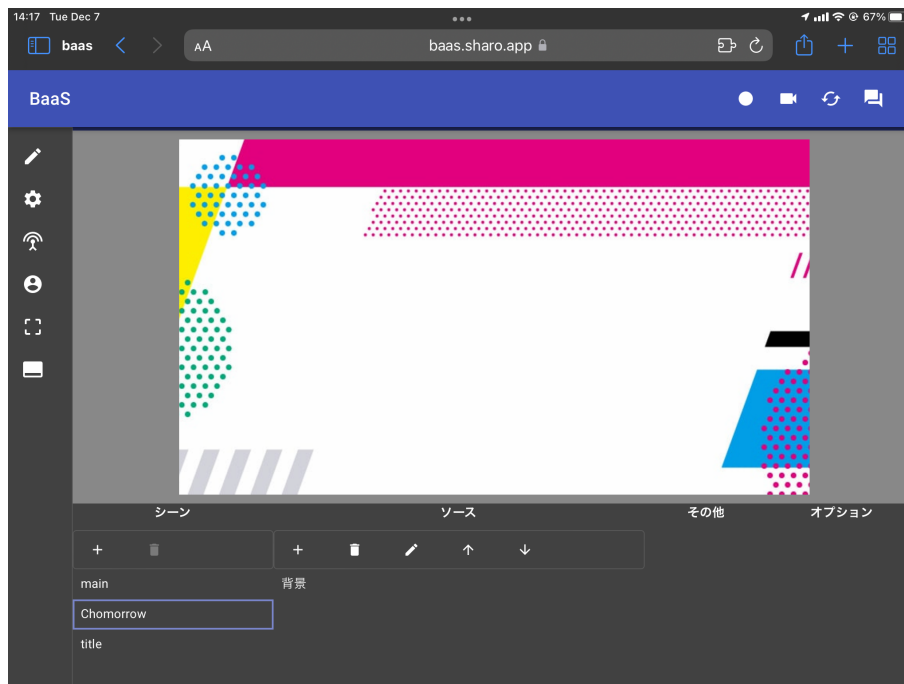


図 4.8 タブレット版 MILC

また、ミキシングソフトウェアの様な複雑な機能を要するアプリケーションは、スマートフォンの様な画面のサイズが小さいデバイスにおいてそのまま UI を構成することは操作面において非常に難しい。1.4 節において定義した「配信することに対する心理的障壁を取り除く」を実現するためには煩雑に見える UI を改善する必要がある。MILC では配信に対する経験の少ない人に対して直感的に操作することができるように、やりたいことベースで操作が行えるよう設計を行った。図 4.10 はソースの配置をドラッグ&ドロップで行う画面である。スマートフォンの場合でもソースをスライド及びピンチで操作することが出来る。

4.4.3 構成

MILC 全体の構成は図 4.11 に示す。サーバは主に 4 つに分かれており、Web サーバ、コントロールサーバ、ストレージサーバ、ミキシングサーバに分かれている。Web サーバはクライアントアプリケーションをホスティングする専用のサーバで

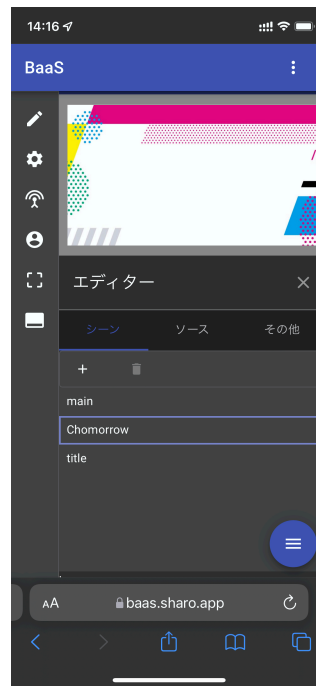


図 4.9 スマートフォン版 MILC

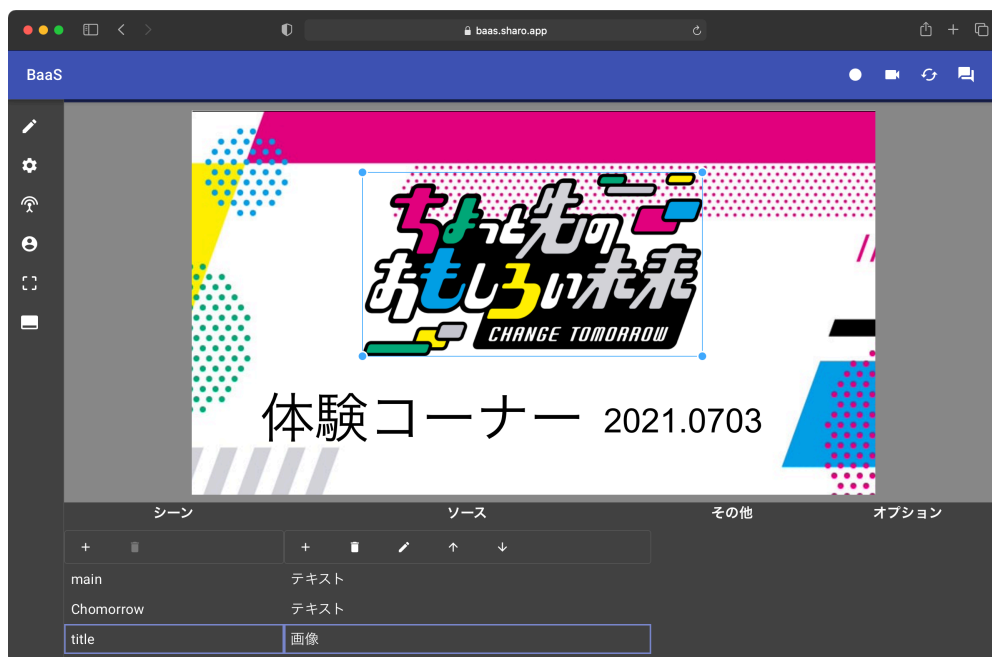


図 4.10 ソース配置の操作

ある。コントロールサーバではユーザ管理，ユーザに応じたミキシングサーバへコマンドを送受信する API の提供を行う。ストレージサーバではミキシングに必要なコンテンツをミキシングサーバへ提供，及びレコーディングしたアーカイブを保存する。ミキシングサーバはユーザ毎にミキシングソフトウェアを起動させ，独立したミキシング環境を提供する。

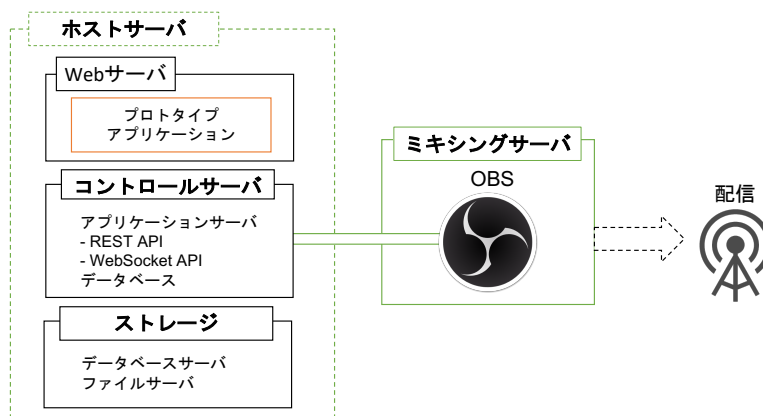


図 4.11 MILC 構成

4.4.4 プログラム

MILC のアプリケーションはクライアントとサーバに分かれている。クライアントサイドのアプリケーションでは，プラットフォームに左右されにくいよう HTML+JavaScript(一部 TypeScript を含む)にてコーディングを行い，ライブラリ群として React と Material-UI を使用した。サーバアプリケーションは，JavaScript でコーディングを行い，ランタイム環境として Node.js，API を提供する手法として https と websocket を使用した。

4.4.5 ネットワーク構成

MILC のために構築したネットワークの構成を図 4.12 に示す。ミキシングサーバを支えるエコシステムは sharo1 及び sharo3 にてサーバアプリケーションを起

動し、ミキシング専用サーバとして sharo5 上に obs-websocket をインストールした OBS を起動している。sharo3 は sharo1 上に VirtualBox にて仮想マシンとして動作している。それぞれのサーバの仕様はそれぞれ sharo1 は表 4.1, sharo3 は表 4.2, sharo5 は表 4.3 に表した。それぞれインストールされているサーバとその役目は以下の通りである。

ファイルサーバ

ミキシングサーバで利用するファイルを保管するサーバである。samba で構成し、ユーザ毎に独立したディスク空間が存在する。主に画像や動画ソースの利用、配信内容をレコーディングする際の保存場所として利用する。使用したサーバアプリケーションは Samba 4.13-Ubuntu である。

STUNTURN サーバ

ミキシングサーバへ映像を送信する際に利用される WebRTC の接続を管理、シグナリングを行うサーバである。主にブラウザ間で P2P 通信を確率させるため、NAT を超えて接続する為の接続情報の交換を行う。サーバアプリケーションは coturn TURN server project による coturn [57] で構成している。

Web サーバ

MILC で使用している Web サーバは 3 種類である。sharo1 上にある Web サーバのアプリケーションは Apache2 を使用しており、MILC の Web アプリケーションを提供する。sharo3 上にある NodeJs の https は、sharo5 上にある OBS を遠隔操作する為の REST API を提供する。sharo3 の websocket サーバは、OBS を遠隔操作及び状態を送受信する ws プロトコルを提供する。リアルタイム性の高いデータの送受信が必要な場合、ws プロトコルを使用する。

ミキシングサーバ

映像をミキシング及びエンコーディングする機能を提供するサーバである。サーバ上に OBS が起動しており、外部から遠隔操作を行うための obs-websocket プラグイン [58] をインストールしている。OBS はオープンソースで利用することができ、また機能や遠隔操作を行うためのドキュメントが充実していることから採用した。

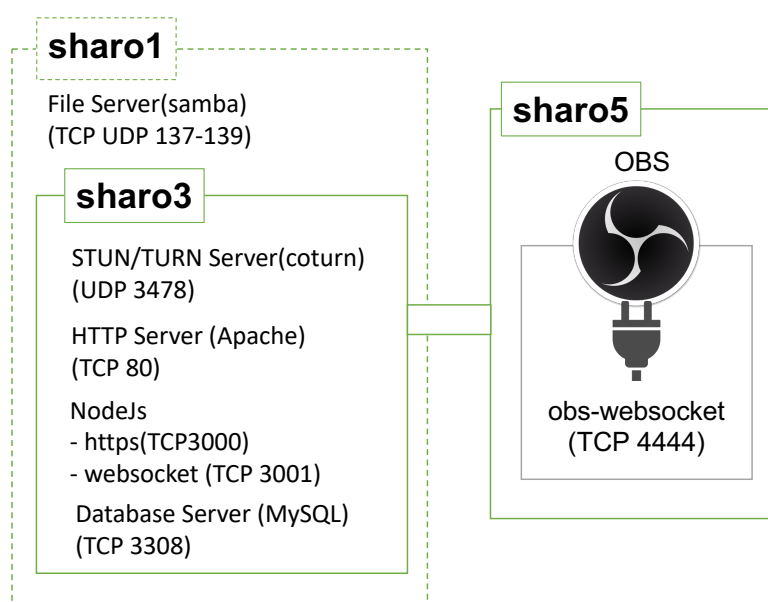


図 4.12 ネットワーク構成

表 4.1 ホストサーバ (sharo1) スペック

OS	Ubuntu 20.04 LTS
プロセッサ	Intel Core i5-8400
メモリ	16GB
ストレージ (SSD)	TOSHIBA THNSNJ51 512GB
ストレージ (HDD)	WDC WD30EFRX-68E 3TB
仮想化環境	VirtualBox

表 4.2 アプリケーションサーバ (sharo3) スペック

OS	Ubuntu 20.04 LTS
プロセッサ	仮想プロセッサ Intel Core i5-8400
メモリ	2GB
ストレージ	VBOX HARDDISK 30GB
ホスト	sharo1

表 4.3 ミキシングサーバ (sharo5) スペック

OS	Windows Server 2019 Datacenter
プロセッサ	Intel Core i7-6700K
メモリ	16GB
グラフィックカード	NVIDIA GeForce GTX 1060 6G
ストレージ	SAMSUNG MZVLW256HEHP-000L7 256GB
OBS バージョン	27.0.1 (64bit)

4.4.6 映像伝送

MILCではWebアプリケーションとして動作するクライアントを提供している。そのためブラウザ上で動作する映像伝送方式を使用する必要がある。しかし現状ではRTSPやNDIといった既存のプロトコルはブラウザ上では使用することができない。そのため、本研究では様々なデバイスにて動作するWebRTCを使用した映像伝送を行った。4.4.7節に詳細を述べた。

4.4.7 レンダリングコンテンツ

カメラやセンサー等の入力によりレンダリングされるコンテンツが変わる動的コンテンツは、そのコンテンツをWebアプリケーションとして作成、それをクライアント及びミキサーに組み込むことで実現した。例として、図4.13に示すように、「位置情報に基づいて現在いる位置を地図上にプロットする」機能を利用する場合、クライアントとミキサー双方にWebアプリケーションで構築した地図

を表示，デバイスはレンダリングに必要なパラメータ（データ）をミキサーに送信し，表示を同期させる．また，コンテンツの内容によってはクライアントからミキサーへパラメータを送る必要のないものも存在する．MILC ではサンプルとして，YouTube Live のコメントを表示するソースの作成も行った．



図 4.13 レンダリングコンテンツ：地図

WebRTC を用いた映像入出力

MILC では，クライアントとサーバ間の映像伝送に WebRTC を用いた．WebRTC はブラウザ上で完結するデータ伝送の仕組みであり，映像や音声も伝送することが可能である．クライアントとミキシングソフトウェア上に，「WebRTC による映像伝送 Web アプリケーションを」クライアントとミキサー上に配置，クライアントとサーバアプリケーションを P2P にて接続し，サーバアプリケーション上で映像を表示する方法を使用している．通信及び表示のモデルを図 4.14 に示す．

4.5. 本章のまとめ

本章では，3 で挙げた課題点を解決するため，Broadcast as a Service のモデルを参考にし，クラウドコンピューティングにより配信環境を提供する手段を提案した．本章にて提案する手法では，配信に関わる要素である映像の入出力やミキサーのコントロールを全て IP 通信を介して操作を行うことで，インターネット上

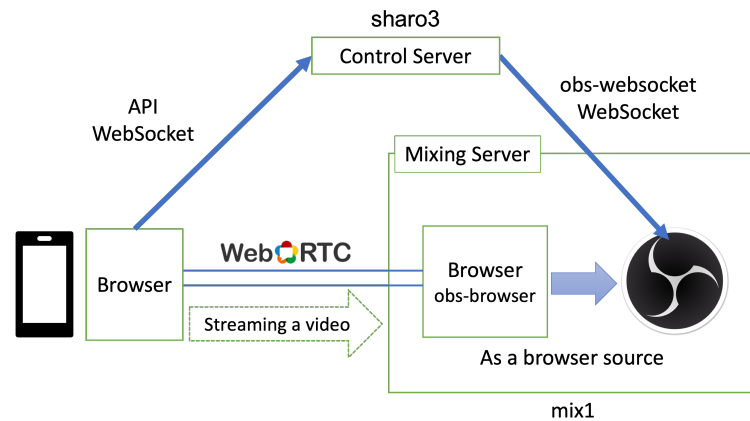


図 4.14 WebRTC を用いた通信のモデル

にて入力から出力まで完結する。ミキシング環境を独立して構築，オンデマンド方式にて環境を提供することにより，複数ユーザが独立・並列して配信を行うことができるため，ユーザ同士で接続を行えば，ユーザ間によるコラボラティブな配信を行うことが可能である。

第 5 章

実験及び評価

5.1. 概要

モバイルデバイスを用いたミキシング環境として，クラウドコンピューティングによるミキシング環境の有効性について評価を行う．その環境を利用して得られる情報の活用手段として，有効性の検証を目的とする．評価実験においては，4にて制作したMILCを実際に参加者に利用してもらい，実際に配信してもらった．また同時に，映像の品質や可用性などをクライアントアプリケーション，サーバアプリケーションともにログを取得し，モバイル回線を利用したクラウドミキシング環境の有用性を検証した．

5.1.1 評価

評価の方法として定量評価及び定性評価を行った．

定量評価

モバイル回線を利用して実際に配信を行う．その際にクライアント及びサーバ側で取得した通信品質，及び映像の品質 (解像度やビットレート等) や鮮明さ (ノイズ) などのログを取得し，それにより従来のコンピュータを利用した方法と比べ，どの様に変化するかを評価する．また，4G回線と5G回線の双方の通信システムにてログを取得し，回線そのものが及ぼす変化についても調査を行った．

定性評価

3章であげた問題点を解決するため、4章にてあげた要件を満たしているか評価する。要件は以下のとおりである。

- モバイルデバイス単体での高度な配信の実現 (4.2.1 節)
- 高品質な配信の実現 (4.2.2 節)

5.2. KMD Forum 実験

5.2.1 実験内容

2021年7月3日に東京ポートシティ竹芝オフィスタワーにて「ちょっと先のおもしろい未来 -Change Tomorrow-」¹が開催された。本実験での展示はBaaSの展示として、Softbank株式会社と共同で行った。展示スペースの構成は図5.1に示す。Softbank株式会社のブースではデジタルアートや映像制作プロダクション向けのBaaS環境の展示され、本研究のブースは個人向けBaaSの体験コーナーとして、4章にて開発したプロトタイプを展示、実際に訪問者に対し同意を得た上で配信を体験してもらい実験を行った。ブースへは10代未満から60代までの男女約300人訪れた。

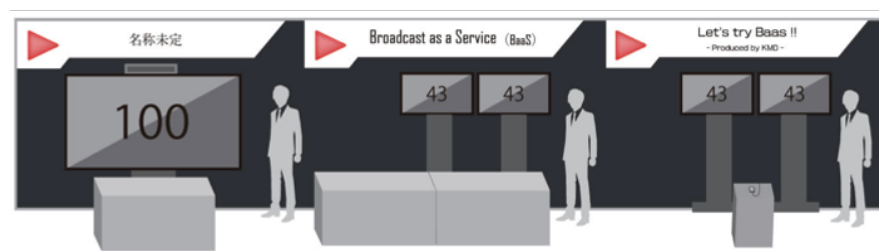


図 5.1 展示構成

1 <https://change-tomorrow.tokyo>

また当日使用した実験の条件については表 5.1 に示す。参加者による実験を行っていない際は 2 台のスマートフォンを利用し、それぞれの映像ソースをミキシングサーバへ入力をした。

表 5.1 秋葉原実験の条件

日時	2021 年 7 月 20 日 11 時より 7 時間
場所	東京ポートシティ 竹芝オフィスタワー 1F
使用デバイス	Sony Xperia Pro 2 台
通信種別	Softbank 5G エリア [43]
コーデック	VP9 + opus
クライアントブラウザ	Chrome for Android 91.0.4472.164
サーバブラウザ	Chromium Embedded Framework 75.1.16+g16a67c4
カメラ解像度	1920x1080@30FPS
ビットレート	可変

5.2.2 実験結果

クライアントからサーバに対しては WebRTC で通信しており、カメラの解像度は 1920x1080 の 30FPS でサーバへ送信する設定であったが、1 時間ほどで過負荷のためサーマルスロットリングが発生した。これにより本来の性能を活かすことができず、自動的にダウンスケールされ、640x360 まで解像度が低下した。またブロックノイズや映像の遅延が目立つようになった。また、2 時間半ほどでバッテリー残量がなくなり、バッテリーを急速に消耗した。ミキシングサーバから YouTube Live サーバに対しては安定して通信しており、モバイルデバイスは交互に使用したが、配信自体は 2 時間 24 分連続で行うことができた。実際に YouTube Live へ



図 5.2 展示光景

配信したスクリーンショットは図 5.3 の通りである。

5.3. 秋葉原実験

5.3.1 実験内容

本実験では 4G/5G 環境による通信や映像の品質を調査するため、4G 回線のスマートフォンと 5G 回線のスマートフォン 2 台を横に並べ、秋葉原エリアを歩いて回った。実験に使用した条件は表 5.2 の通りである。また実験で歩いたルートについては図 5.4²の通りであり、安全に実験を行うため、スマートフォンの画面は斜め下に向け画面は見えていないが、2 台のスマートフォンが可能な限り同じ場所を撮影するように心がけた。

2 電子地形図 25000(国土地理院) [42] を加工して作成した。

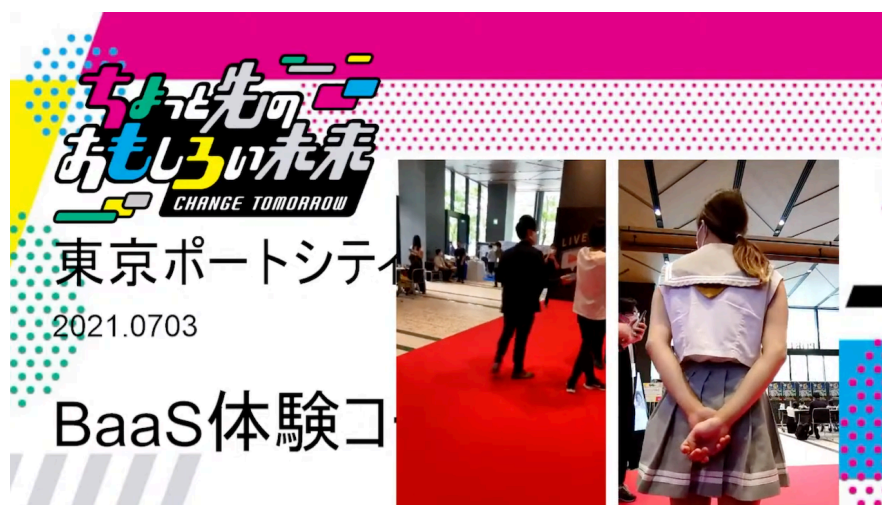


図 5.3 配信画面

5.3.2 実験結果

本調査では WebRTC において最もメジャーな VP9 コーデックを用いて長時間配信した際の通信品質を 4G, 5G 通信下で比較した。本実験の条件は 5.2 の通りである。

転送量と解像度の推移を解析すると 5.6 のようになった。このデータは開始から 10 分間のデータである。起動した直後は共に大きい解像度での映像伝送を試みたが、5G では可変ビットレートで動作し、安定した映像伝送ができていたのに対し、4G では最低限の固定ビットレートに引き下げられた。アップロードの通信速度最大値は 5G 通信 103Mbps、4G 通信 15Mbps であるが³ [59]、実際は 4G 通信下においてはビットレートの高くなる高解像度の伝送は安定した通信を行うことはできなかった。5G/4G エリアをまたいだ配信を行ったところ、通信種別が切り替わるたびに WebRTC の通信は切断された。現状は 5G/4G の切り替えはシームレスには行かず、切断されるたびに再接続が必要になることから、配信スタイルに応じてはネットワーク種別の固定、ないしは WebRTC の再接続の高速化を検討

3 ベストエフォートな回線であるため、通信速度の最大値で通信を行うことは現実的には不可能である。

表 5.2 秋葉原調査の条件

日時	2021年7月20日 17時55分より18分間
調査エリア	秋葉原 中央通り沿い
使用デバイス	Sony Xperia Pro 2台
5G デバイスの通信種別	Softbank 5G [43]
4G デバイスの通信種別	Softbank 4G
コーデック	VP9 + opus
クライアントブラウザ	Chrome for Android 91.0.4472.164
サーバブラウザ	Chromium Embedded Framework 75.1.16+g16a67c4
カメラ解像度	1920x1080@30FPS
ビットレート	可変

する必要がある。

5.4. 定性評価

4.2 節にて述べた要件に対して、本研究で提案した MILC が実現した機能の定性的な評価は以下の通りである。

モバイルデバイス単体での高度な配信の実現

KMD Forum 実験及び秋葉原実験の結果より、クラウドコンピューティングを利用することにより、モバイルデバイスでも高度な配信を行うことは可能であった。モバイルデバイスで使用しているリソースはカメラの映像のエンコード、ストリーム送信が大半であり、負荷が大きかったためバッテリーの消耗は大きかった。デバイスの状態をモニタリングし、細かくコントロールすることで長時間の配信も可能であった。



図 5.4 秋葉原実験 実験範囲

高品質な配信の実現

従来のモバイルデバイスからの配信と比較して、本実験ではテキストや画像、複数のカメラを入力及びミキシングした配信を行った。またこの配信をデバイスは切り替えつつも2時間24分行うことができ、現実的に使用できる環境を構築することが可能であった。また、4G及び5G通信を利用した映像伝送比較の実験において、5G通信の方が余裕のある通信が実現できることが分かった。これにより安定した高品質な映像伝送を行うには最大15Mbpsの回線では不十分であり、より高い要件が要求されることが分かる。

5.5. 定量評価

秋葉原調査において得られたログより、5G通信と4G通信とでは、5G通信の方がより高品質な通信が行えることが分かった。しかしハードウェアの高負荷によるサーマルスロットリングが見られたため、映像品質に及ぼす影響が通信品質だけではなく、ハードウェアによる安定した処理性能も重要なポイントの一つと



図 5.5 秋葉原調査

言える。より精度の高い評価を行うにはハードウェアの状態をモニタリングしつつ、コーデックの選択や映像品質のコントロールを行う必要があり、今後の課題として残される。

5.6. 本章のまとめ

本章では、4章にて提示したクラウドコンピューティングによる配信環境のMILCを用いて、KMD Forum及び秋葉原にて実験を行った。定性評価では、モバイル単体による映像や文字、画像を組み合わせたミキシング配信を行うことができ、YouTube Liveへ出力の出力は2時間24分と長時間行うことが確認できた。定量評価では、同一デバイス・同一環境にて4G通信と5G通信で映像伝送比較実験を行ったところ、5G通信がより大容量・高帯域により余裕のある通信を行うことができ、高品質な映像伝送を行うことが可能であった。このことより、本研究におけるクラウドコンピューティングによる配信環境の提供手法は、従来の手法と比べ、準備する機材を減らし、より幅広いデバイスから高度な配信を行うことができ、新たな配信環境としての有効性を確認した。

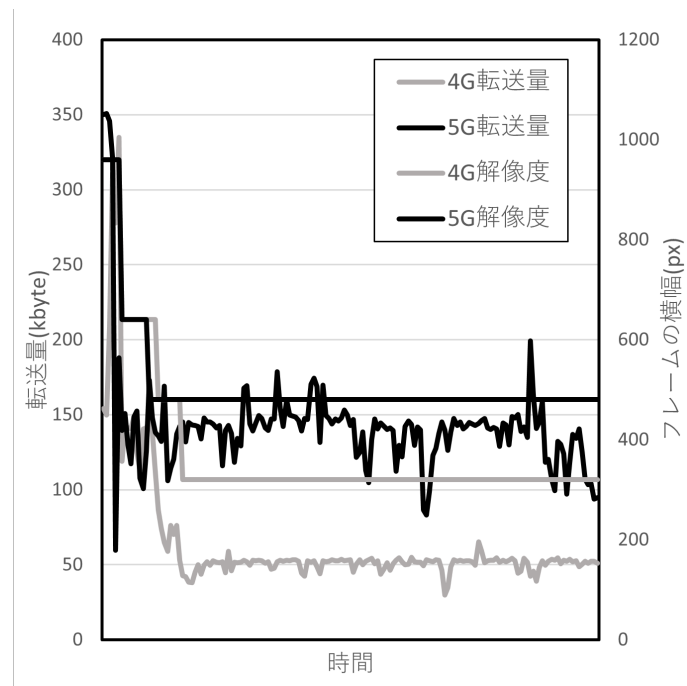


図 5.6 転送量と解像度の推移

第 6 章

結 論

6.1. 本研究において実現した事項

6.1.1 技術的要件における解決

本研究におけるプロトタイプにより、従来では必要だとされていた高性能なコンピュータはネットワークを通じてクラウドコンピューティングにて提供することができ、またカメラ等の入力においてもそのままスマートフォン等のモバイルデバイスのもをストリーム転送することで使用することが可能であることが分かった。また、4G/5G 実験において、次世代の移動通信システムである 5G の大容量通信により、4G と比べて高品質な映像転送を安定して行うことができることが分かった。しかし、スマートフォンでは大容量の送信をモバイルデバイスから長時間することは想定されておらず、環境に応じて通信をコントロールしなかった場合、ハードウェアのサーマルスロットリングにより処理性能が自動的に制限され、映像品質が低下することが分かった。

映像転送に限らず、GPS などセンサーのデータをサーバへ送信し、動的なコンテンツをサーバ上でレンダリングすることにより、よりインタラクティブな表現をアプリケーション上で行うことが可能であると分かった。またレンダリングするコンテンツはミキシングサーバで独立して動作するため、必要に応じてユーザ自身がアプリケーションを定義することでより拡張性の高い機能を提供することが分かった。ミキシングサーバ自体も仮想マシンにより動作するため、ユーザ毎に独立した配信環境を提供することが出来る。同時に仮想マシンで動作をコントロールすることが出来るため、仮想マシン同士でネットワークを構築すると、遠隔地にいつつコラボラティブなコンテンツを作ることが出来る。

6.1.2 心理的障壁の解決

本研究におけるプロトタイプでは、配信の経験の少ないユーザに対して気軽に使ってもらえるよう、ユーザインターフェースを簡素化し、直感的に操作することができるよう構築を行った。実際に一般の方が訪れるイベントにおいて、コンテンツ制作プロダクション向け BaaS と横並びに展示したところ、何をすることができるのか、どの様に使えばいいのかわかりやすいと好意的な意見を得ることができた。

6.2. 本研究の貢献

本研究では、従来のリアルタイム型コンテンツ配信に対して、1) 場所や環境を問わず配信を行うことが出来る配信環境のモデルを提案し、2) モデルに基づいたクラウドコンピューティングを利用した配信環境 (MILC) を構築し、新たな配信環境を実現した。MILC は 3.5 節の Broadcast as a Service のモデルを利用し、配信環境を支えるシステムをエコシステムとして動作させ、ミキシングを提供するサーバを独立して動作させることで、コラボラティブな配信環境を独立・並列に動作させた。MILC のモデルは図 4.1 に示した。MILC により、1) モバイルデバイス単体での高度な配信の実現、2) 高品質な配信の実現の要件を定性評価及び定量評価にて評価を行い、新たな配信環境としての有効性を確認した。

本研究による MILC を活用することにより、PC など高性能なコンピュータを有していないユーザでも PC と同様のミキシングを行えるようにし、環境によって表現の幅が大きく制限されることのない配信を可能とした。また、MILC でサポートしている映像伝送プロトコルは一般的に使用されているものであり、現在の主要な配信プラットフォームでは概ね MILC が有効的に使用できる。

ユーザジェネレート型コンテンツ (UGC) は同じ趣味や志のユーザ同士が作り上げるコンテンツであり、ユーザ間が相互に作用しつつ制作できる環境は重要である。クラウド上の環境及びリソースは共有して利用することが可能であり、複数ユーザによって 1 配信を作ることや、複数配信を 1 配信として制作するコラボ

ティブな配信を実現でき、「ユーザ配信」のユーザジェネレート型コンテンツとしての価値を高めることができた。

6.3. 今後の課題

6.3.1 配信環境における課題

本研究では、モバイルデバイスにおける配信環境を実現するためのクラウドコンピューティングを活用したミキシング環境について提案を行い、それを評価した。今後、さらなる技術の進化に伴い、通信環境やハードウェアの変化やあり方も変化していくと予想される。特に、モバイルデバイスの小型化やパーソナライズ化が進むにつれ、よりセンサーなどのデバイスが人と密接に関わるようになり、デバイスそのものが単純な入力機器、またはコントローラとしての役割に変化していく。インタラクティブな配信環境を提供するには、デバイスから送信されたパラメータに基づいて、サーバ上でレンダリングする仕組みをプラットフォーム化し、よりオープンに利用することができる環境構築を行う必要があると考える。本研究ではスマートフォンに着目し、ミキシング環境をクラウドコンピューティング化することに着目を行ったが、今後はその多様化、及びサーバ内で扱うデータやレンダリングコンテンツのあり方について議論が必要となってくる。

6.3.2 技術的要件における課題

環境構築するにあたって技術的に重要視したのはネットワークとデバイスである。ネットワークにおいては5Gの高速通信及び大容量通信によりボトルネックとなる問題は解消したと言える。しかしミキシングサーバへ入力を行うスマートフォンにおいて、カメラから入力した映像を単純にエンコード及びストリーム送信するだけであったとしても、デバイスが熱を持ちサーマルスロットリングによる性能低下が起こされた。今後はハードウェアなど外的要因による自動的な品質の調整を行うマネジメント機構が必要となる。また、より身近なデバイス、例えばスマートウォッチやIoT機器と配信環境との連携は、インタラクティブな配信を行う手

段として非常に重要と言える。デバイスに依存しない抽象化されたシステム間連携を行うため、一貫された通信方式(プロトコル等)の議論が必要となる。

謝 辞

本研究を進めていくにあたり助言や御指導をしていただきました，すべての皆様に感謝いたします。

本論文の主旨導教員であり，研究を進めていく上で多くの助言や御指導をしていただきました。また，多くの優しい励ましのお言葉を頂戴し，研究をここまで続けていくことができました。慶應義塾大学大学院メディアデザイン研究科の杉浦一徳教授に最大限の感謝をいたします。

研究の方向性や細かな点において助言や御指導をしていただきました，慶應義塾大学大学院メディアデザイン研究科の砂原秀樹教授に心から感謝いたします。

研究指導や論文執筆において様々な助言と御指導をいただきました，慶應義塾大学大学院メディアデザイン研究科の南澤孝太教授に心から感謝いたします。

本研究を進めていく上で重要な調査及び検証にて協力，御助言を頂きました。Softbank 株式会社の湧川隆次氏，船吉秀人氏，時村晋太郎氏，長内大茂氏に心から感謝いたします。

本論文の実験や検証において様々な御指摘，御指導や助言をいただきました，WIDE Project のみなさま，情報処理学会マルチメディア通信と分散処理研究会のみなさまに深く感謝をいたします。

最後にここまで24年間の長い間支えていただきました父 山下仁志，母 山下陽子に最大級の感謝をいたします。多くの励ましの御言葉や支援をいただきました祖母 和田和子，祖父 和田久司に深く感謝いたします。

参 考 文 献

- [1] YouTube LLC. Youtube. URL: <https://youtube.com> [2021/12 アクセス].
- [2] DWANGO Co., Ltd. ニコニコ動画. URL: <https://www.nicovideo.jp/> [2021/12 アクセス].
- [3] Microsoft. Skype. URL: <https://www.skype.com/> [2021/12 アクセス].
- [4] OISEYER Inc. Keyholetv. URL: <https://www.oiseyer.com/index-j.php> [2021/12 アクセス].
- [5] YouTube LLC. Youtube live. URL: <https://www.youtube.com/live> [2021/12 アクセス].
- [6] DWANGO Co., Ltd. ニコニコ生放送. URL: <https://live.nicovideo.jp> [2021/12 アクセス].
- [7] Macromill. ライブ配信に関する調査, 10 2017.
- [8] YouTube LLC. Create a live stream on mobile. URL: <https://support.google.com/youtube/answer/9228390> [2021/08 アクセス].
- [9] St 2082-1:2015 - smpte standard - 12 gb/s signal/data serial interface — electrical. *ST 2082-1:2015*, pp. 1–14, 2015. doi:10.5594/SMPTE.ST2082-1.2015.
- [10] Anup Rao, Rob Lanphier, and Henning Schulzrinne. Real Time Streaming Protocol (RTSP). RFC 2326, April 1998. URL: <https://rfc-editor.org/rfc/rfc2326.txt>, doi:10.17487/RFC2326.

- [11] Henning Schulzrinne, Anup Rao, Rob Lanphier, Magnus Westerlund, and Martin Stiemerling. Real-Time Streaming Protocol Version 2.0. RFC 7826, December 2016. URL: <https://rfc-editor.org/rfc/rfc7826.txt>, doi: 10.17487/RFC7826.
- [12] NewTek Inc. Ndi. URL: <https://www.ndi.tv> [2021/11 アクセス].
- [13] Justin Uberti, Cullen Jennings, and Eric Rescorla. JavaScript Session Establishment Protocol (JSEP). RFC 8829, January 2021. URL: <https://rfc-editor.org/rfc/rfc8829.txt>, doi:10.17487/RFC8829.
- [14] Information technology — Coding of audio-visual objects. ISO/IEC 14496, December 2020.
- [15] Information technology — generic coding of moving pictures and associated audio information. ISO/IEC 13818. ICS : 35.040.40, June 2019.
- [16] Information technology — high efficiency coding and media delivery in heterogeneous environments. ISO/IEC 23008, August 2020.
- [17] Adrian Grange, Peter de Rivaz, and Jonathan Hunt. VP9 Bitstream & Decoding Process Specification. URL: <https://www.webmproject.org/vp9/> [2021/12 アクセス].
- [18] Peter de Rivaz and Jack Haughton. AV1 Bitstream & Decoding Process Specification. URL: <https://github.com/AOMediaCodec/av1-spec/releases/tag/v1.0.0> [2020/12 アクセス].
- [19] Google. Av1 encoder for webrtc, new origin trials, and more. URL: <https://blog.chromium.org/2021/03/chrome-90-beta-av1-encoder-for-webrtc.html> [2021/08 アクセス].
- [20] 桜井茂男. 内発的動機づけに及ぼす言語的報酬と物質的報酬の影響の比較. 教育心理学研究, Vol. 32, No. 4, pp. 286–295, 1984. doi:10.5926/jjep1953.32.4_286.

- [21] AbemaTV, Inc. Abema. URL: <https://abema.tv> [2021/12 アクセス].
- [22] DAZN. Dazn. URL: <https://www.dazn.com> [2021/12 アクセス].
- [23] 17LIVE Japan Inc. 17live. URL: <https://jp.17.live> [2021/12 アクセス].
- [24] Meta. Instagram. URL: <https://www.instagram.com> [2021/12 アクセス].
- [25] Moi Corp. ツイキャス. URL: <https://twitcasting.tv> [2021/12 アクセス].
- [26] TikTok. Tiktok. URL: <https://www.tiktok.com> [2021-12 アクセス].
- [27] LINE Corporation. Line live. URL: <https://live.line.me> [2021/12 アクセス].
- [28] DWANGO Co., Ltd. ニコニ・コモンズ. URL: <https://commons.nicovideo.jp> [2021/12 アクセス].
- [29] Netflix, Inc. Netflix. URL: <https://www.netflix.com/> [2021/12 アクセス].
- [30] NTT DOCOMO. d アニメ. URL: <https://anime.dmkt-sp.jp/> [2021/12 アクセス].
- [31] 三菱 UFJ リサーチ&コンサルティング. ライブ配信サービス（投げ銭等）の動向整理. URL: https://www.caa.go.jp/policies/policy/consumer_policy/policy_coordination/internet_committee/pdf/internet_committee_190117_0002.pdf [2021/08 アクセス].
- [32] 三島和宏. 自律分散型コンテンツ情報管理アーキテクチャ. 博士論文, 2012.
- [33] TVU Networks. Tvu networks. URL: <https://www.tvunetworks.com> [2021/12 アクセス].
- [34] LiveU Inc. Liveu. URL: <https://www.liveu.tv> [2021/12 アクセス].
- [35] スターコミュニケーションズ株式会社. Tvu pack について. URL: <https://starcom.co.jp/tvu/> [2021/12 アクセス].

- [36] Warchamp7. Obs studio. URL: <https://obsproject.com> [2021/11 アクセス].
- [37] SplitmediaLabs. Xsplit. URL: <https://www.xsplit.com> [2021/11 アクセス].
- [38] おかゆう. ゆかりねっと.exe. URL: <http://www.okayulu.moe> [2021/12 アクセス].
- [39] みちあき. 棒読みちゃん. URL: <https://chi.usamimi.info/Program/Application/BouyomiChan/> [2021/12 アクセス].
- [40] Ryu. マルチコメントビューア. URL: <https://ryu-s.github.io/app/multicommentviewer> [2021/12 アクセス].
- [41] moro. Niconama Comment Viewer. URL: <https://www.posite-c.com/application/ncv/> [2021/12 アクセス].
- [42] 国土地理院. 電子地形図 (25000). URL: <https://maps.gsi.go.jp> [2021/11 アクセス].
- [43] SoftBank Corp. サービスエリアマップ. URL: <https://www.softbank.jp/mobile/network/area/map/> [2021/08 アクセス].
- [44] Alice Dev Team. Periscope hd. URL: <https://apps.apple.com/us/app/periscope-hd-h-264-rtsp-cam/id1095600218> [2021/08 アクセス].
- [45] Palakis. obs-ndi - github. URL: <https://github.com/Palakis/obs-ndi/releases> [2021/08 アクセス].
- [46] NewTek. Ndi discovery and registration. URL: <https://support.newtek.com/hc/en-us/articles/218109477-NDI-Discovery-and-Registration> [2021/08 アクセス].
- [47] Mark Gilbert. Ndicam. URL: <https://apps.apple.com/jp/app/ndicam/id1074065391> [2021/08 アクセス].

- [48] スターコミュニケーションズ株式会社. Tvu one レンタル. URL: <https://starcom.co.jp/tvu/price.html> [2021/08 アクセス].
- [49] TVU Networks Corporation. Tvu anywhere pro. URL: <https://apps.apple.com/jp/app/tvu-anywhere-pro/id883099579> [2021/08 アクセス].
- [50] Teradek. Airmix solo. URL: <https://apps.apple.com/jp/app/airmix-solo/id1051147032> [2021/12 アクセス].
- [51] Softvelum LLC. Larix broadcast. URL: https://play.google.com/store/apps/details?id=com.wmspanel.larix_broadcaster [2021/12 アクセス].
- [52] bmc. Broadcast as a service. URL: <https://www.bmcuk.tv/baas> [2021/08 アクセス].
- [53] singular.live. Singular.live. URL: <https://www.singular.live> [2021/08 アクセス].
- [54] LTN Global Communications, Inc. Ltn. URL: <https://ltnglobal.com> [2021/08 アクセス].
- [55] StreamYard, Inc. Streamyard. URL: <https://streamyard.com> [2021/12 アクセス].
- [56] 総務省. 通信利用動向調査. 情報通信統計データベース, 2019.
- [57] GitHub, Inc. coturn/coturn - github. URL: <https://github.com/coturn/coturn> [2021/12 アクセス].
- [58] Palakis. obs-websocket - github. URL: <https://github.com/Palakis/obs-websocket> [2021/08 アクセス].
- [59] Softbank. [softbank 5g] softbank 4g、softbank 3g との違いを教えてください。 URL: <https://www.softbank.jp/support/faq/view/10993> [2021/08 アクセス].