分散アーカイブ環境において
コンテンツ提供制御を行うソフトウェアの
設計と実装

政策メディア研究科 89931614 _____

平成 13 年 1 月 16 日

Abstract

近年ストリーミング技術を用いたサービスが注目されている。そのようなサービスで提供される動画コンテンツの作成は、今までのテキストや画像主体のコンテンツより作成労力が大きい。また、配布環境の構築に相応の労力が必要となる。そのため、コンテンツの作成者 (コンテンツホルダー) とコンテンツの配布者 (シンジケーション) がそれぞれ専業化する方向にある。

本研究はコンテンツホルダーとシンジケーションが独立した組織である環境において、コンテンツホルダーによるコンテンツ提供ポリシを実現することを目的とする。コンテンツ提供ポリシの設定においては、通常の視聴者属性による静的な視聴可否判断だけでなく、諸条件の変化に応じて動的な視聴可否を決定するような提供ポリシの設定の可能性についても考慮する。

現状では、動的な視聴可否を行えるような、柔軟なコンテンツ提供ポリシ設定を実現する方法は、ストリーミングコンテンツのサーバ実装に依存するが、これは今後の技術の発展や用途の違いにより使用する技術を使い分ける場合に対応できない。このため本研究においては、柔軟なコンテンツ提供ポリシをサーバ実装に依存せずに実現し、同時にコンテンツの拡散の防止に配慮した構造について検討した。

柔軟なコンテンツ提供ポリシを実現するためには、視聴者の認証と過去の視聴履歴の取得をする必要がある。これらの要求を満たすため、本研究においてはストリーミングコンテンツサーバがファイルシステムへアクセスする際に、視聴者の認証とコンテンツ提供ポリシ設定を行うシステムを検討の上、設計・実装した。

本システムを実現することによって、コンテンツホルダーとシンジケーションが独立した組織において運営されている場合であっても、現存するサーバ実装に因われずコンテンツ提供ポリシを設定できることが実証できた。

目 次

1 序論 **1**
1.1 本研究の目的 1
1.2 用語の定義 2
1.3 本論文の構成 3

2 背景 **4**
2.1 インターネットメディアの変遷 4
2.1.1 ストリーミング技術の特徴 4
2.1.2 ストリーミング技術の需要 5
2.2 コンテンツホルダーとシンジケーションの分離 7
2.3 コンテンツ提供ポリシの設定 8
2.4 現状の問題点 8
2.4.1 分散してしまう視聴ログ 9
2.4.2 コンテンツ提供ポリシ表現の柔軟性欠如 9
2.4.3 コンテンツ拡散の危険性 9
2.4.4 認証運用のネットワークメディア依存性 10
2.4.5 現状の問題点のまとめ 11

3 関連研究 **12**
3.1 複数組織で提供するもの 12
3.1.1 mirror 12
3.1.2 分散ファイルシステム 13
3.1.3 RealProxy 14
3.2 柔軟なコンテンツ提供ポリシを設定できるもの 14
3.2.1 Apache において RDBMS を利用するアクセスコントロール 14
3.2.2 RealServer 認証機能 15
3.2.3 Kerberos 15
3.2.4 WMRM(Windows Media Rights Manager) 16

3.3 関連研究の比較 17

4 概要 **19**
4.1 研究の概要 19
4.2 本研究の特徴と期待される成果 20

5 構成 **21**
5.1 構成要素 21
5.2 コンテンツ提供ポリシの種類 23
5.2.1 コンテンツホルダーとシンジケーション間 23
5.2.2 シンジケーションと受講者間 24
5.2.3 コンテンツホルダーと受講者間 25
5.3 コンテンツ提供ポリシ設定 26
5.3.1 ポリシ設定の種類 26
5.3.2 ポリシ設定に必要な情報 27

6 モデルの設計 **29**
6.1 モデル設計の前提 29
6.1.1 既存ソフトウェアの使用 29
6.1.2 URI 指定に Web システムを使用する 30
6.2 要求上の問題 31
6.2.1 必要な情報 31
6.2.2 視聴者の本人性確認 31
6.2.3 なりすましやコンテンツ再送の防止 32
6.3 設計の方向性 32
6.3.1 ポートフックを利用した実装案 32
6.3.2 ファイルシステムとしての実装案 34
6.3.3 本モデルでの設計 35
6.4 モデルの構成 35
6.5 コンテンツ取得の手順 36
6.5.1 URI 作成プロセス 38
6.5.2 ストリーミングコンテンツ提供プロセス 38
6.5.3 セッション ID の有効性 39
6.6 ソフトウェア 39
6.6.1 コンテンツホルダーホストで動作するマスター認証サーバ 40
6.6.2 シンジケーションで動作するコンテンツサーバ 40
6.6.3 シンジケーションで動作するファイルシステム 41
6.6.4 コンテンツホルダーとシンジケーション間の通信 . . . 41
6.6.5 シンジケーションにおける動的ページ生成機構 41

6.7 問題点の解決 41
6.7.1 視聴ログの収集 42
6.7.2 メディアに依存しない認証機構 42
6.7.3 コンテンツの拡散抑制 42
6.7.4 コンテンツ提供ポリシ表現の柔軟性確保 43

7 実証ソフトウェアの設計と実装 44
7.1 システムの構成 44
7.2 実装環境 45
7.3 マスター認証サーバの実装 45
7.3.1 ネットワーク通信部 46
7.3.2 データベースインタフェース 46
7.3.3 暗号系 46
7.4 シンジケーションファイルシステム 46
7.4.1 ファイルシステムデーモン 47
7.4.2 セッションマネージャ 47
7.4.3 データベースインタフェース 48
7.5 動的ページ生成機構 48
7.6 通信プロトコル 48
7.7 ファイル名 49
7.8 ファイルアクセスに対する応答 49

8 評価 51
8.1 他研究との比較 51
8.2 コンテンツ提供ポリシ表現の柔軟性欠如への対応 52
8.3 分散してしまう視聴ログへの対応 52
8.4 コンテンツの拡散への対応 53
8.5 認証運用のメディア依存性への対応 53
8.6 問題点充足のまとめ 54

9 結論 55
9.1 本研究のまとめ 55
9.2 今後の研究課題 55

10 謝辞 57

11 付録 A: 通信プロトコル 58
11.1 プロトコル共通ヘッダ 58
11.2 CREATE_URI_R 時の通信内容 58
11.3 CREATE_URI_A 時の通信内容 60

11.4 OPEN_SESSION_R 時の通信内容 60
11.5 OPEN_SESSION_A 時の通信内容 60
11.6 CONTENTS_READ_R 時の通信内容 61
11.7 CONTENTS_READ_A 時の通信内容 61
11.8 エラーコード 62

12 付録 B: データベーステーブル **63**
12.1 データベーステーブル：コンテンツホルダー 63
12.1.1 シンジケーションリストテーブル 63
12.1.2 コンテンツの情報クラス 65
12.1.3 コンテンツ提供ポリシ選択クラス 65
12.2 データベーステーブル：シンジケーション 66
12.2.1 コンテンツホルダーリスト 66
12.2.2 視聴ログ 67
12.2.3 コンテンツキャッシュ 67

図 目 次

2.1 RealServer によるコンテンツファイルアクセス順 6

3.1 分散ファイルシステム 13
3.2 Kerberos 概要 16

5.1 構成要素関係図 21

6.1 ポートフック案 33
6.2 ファイルシステム案 34
6.3 本システム概要 36
6.4 コンテンツ取得手順シーケンス 37
6.5 システムの構成 40

7.1 システムの構成 44
7.2 シンジケーションファイルシステムの構成 47
7.3 ファイル操作時のファイルツリー 50

表 目 次

3.1 関連研究の比較 18

6.1 ポートフック案とファイルシステム案の比較 35

7.1 実装環境 45
7.2 URI 中のファイル名構成 49

8.1 問題点充足に関する比較 51

11.1 プロトコル共通ヘッダ 58
11.2 通信段階識別子 59
11.3 CREATE_URI_R 時のフォーマット 59
11.4 OPEN_SESSION_R 時のフォーマット 60
11.5 OPEN_SESSION_A 時のフォーマット 61
11.6 CONTENTS_READ_R 時のフォーマット 61
11.7 CONTENTS_READ_A 時のフォーマット 61
11.8 プロトコル：エラーコード 62

12.1 コンテンツホルダー：シンジケーションリストテーブル 63
12.2 コンテンツホルダー：視聴ログクラス 64
12.3 コンテンツホルダー：コンテンツの情報クラス 64
12.4 コンテンツホルダー：提供ポリシ選択クラス 65
12.5 コンテンツホルダー：コンテンツ提供ポリシ記述クラス 65
12.6 コンテンツホルダー：契約記述クラス 66
12.7 シンジケーション：コンテンツホルダーリストクラス 66
12.8 シンジケーション：視聴者リストクラス 66
12.9 シンジケーション：視聴ログクラス 67
12.10シンジケーション：コンテンツキャッシュクラス 67

Chapter 1

序論

今までのインターネットにおいては、著作権の権利関係に関して、現実世界での利害関係とインターネット上での利害関係の相関が希薄な場合がほとんどであった。

これは、インターネットは情報を広く共有する目的で発展して来ており、ビジネスシーンでの利用や遠隔教育の媒体としての利用等、権利関係が重視されるような利用法は主眼としていなかったためである。

しかし、インターネットがビジネスや教育等の目的で利用されるようになって来た現在、コンテンツに対する著作頒布権等の権利を主張することを前提としたシステムが構築される機会が増えて来た。これは例えばインターネット放送局やインターネット上でビデオ オン デマンドを実現するシステム、イントラネット上で社内向けコンテンツ配信をするシステム等を構築する場合などが該当する。上記のようなシステムを構築するために、著作頒布権の反映を確実に遂行できる仕組みを用意しなくてはならない。

1.1 本研究の目的

本研究はコンテンツホルダーとシンジケーションが別の組織である場合に、コンテンツホルダーによるコンテンツ提供ポリシを実現することを目的とする。この際に、これを実現する上で特定のストリーミングメディア技術に依存する必要が無いこと、コンテンツの意図せざる拡散を避けることができることが要求される。また、コンテンツ提供ポリシを実現するための情報として、視聴者が該当視聴者であるかどうかを認証すること、視聴ログがコンテンツホルダーに残ることが求められる。

1.2 用語の定義

ここに本論文中に用いる用語の定義を行う。

コンテンツ 特に断りが無い場合、本研究においては視聴者がメディアクライアントを用いて視聴する動画番組の内、サーバに保管されており、ストリーミング技術を用いて配布することを意図したものを指す。

アーカイブ 視聴者からの要求に応じて提供できるような形にしてあるコンテンツデータであり、サーバ上に保管されている。また、時として保管するための保存形式を指す。アーカイブとして保管されているものは、ライブ形式で提供されるものと比べて、視聴者がコンテンツ視聴をする際に時間的な自由度がある。

コンテンツホルダー (1 次提供者) コンテンツの著作頒布権を持つもの、もしくはその執行を委託されたもの。遠隔教育環境においては、講義を行う者の所属校を意味する。

シンジケーション (2 次提供者) コンテンツホルダーからの委託等により、コンテンツの配布を受け持つもの。シンジケーションは、コンテンツホルダーとの契約に応じてシンジケーションが持つアーカイブから、もしくはコンテンツホルダーが持つアーカイブからコンテンツを取得し、視聴者のメディアクライアントに対してサービスを行う。遠隔教育環境においては、自校の単位として組み入れるために、受講者に対して他校の講義を提供する者を意味する。

メディアクライアント 視聴者がストリーミングコンテンツを視聴するために用いるソフトウェア。RealPlayer, WindowsMediaPlayer, MotionViewer など、ストリーミングメディア技術に応じて多様な種類のメディアクライアントが存在する。

コンテンツサーバ シンジケーションが視聴者に対してコンテンツストリーミングをサービスをする為のソフトウェア、もしくはそのようなソフトウェアが動作しているインターネットホストを指す。

サービス クライアント / サーバ型システムにおける、サーバによる情報提供のことを指す。

ストリーミングメディア 本研究においては、ストリーミングコンテンツの圧縮保管方法と送受信する際の形式を指す。メディアの種類としてはRealMedia や QualityMotion などが挙げられる。

動的ページ生成機構 ウェブサービスにおいて、動的にページを生成するために構成されたプログラムやインタフェースを指す。例えば、CGI(Common Gateway Interface) や SSI(Server Side Include), PHP, JSP(Java Server Pages) などの仕組み一般を表す。

ホスト 本研究においては、インターネット上において、あるサービスを行うために運用されている 1 台のコンピュータ、もしくはクラスタリング構成された複数台のコンピュータ群を指す。

ポリシ 本研究においては、コンテンツを提供する際に、コンテンツホルダーによって設定される視聴条件を指す。ポリシは、視聴者に対する一元的な視聴可否だけに留まらず、例えば遠隔教育環境において、ある講義を受講済であることを条件に特定の講義の視聴を許可するというような、多元的な条件付けを行った視聴可否の選択を含む。また、コンテンツホルダーとシンジケーション間の取り決めに従ってシンジケーションにおいてコンテンツの保存を許可するかどうかや、時間帯制限・回数制限等の配信に関する条件付けも、ポリシという用語に含む。

1.3 本論文の構成

本論文は以下のような構成である。

第 2 章で本研究の背景となる事柄について述べ、第 3 章では、本研究に関連するストリーミング技術と認証技術についての関連研究を説明する。第 4 章で本研究の概要として、他の技術と比較した本研究の特徴とそれによって期待される成果を述べる。第 5 章では本研究を検討する上で関係する各構成に付いて整理し、第 6 章で問題点の解決を本研究を実現するコンテンツ提供制御を行うためのモデルを設計する。第 7 章で、このモデルが有効かどうか実証するために作成するソフトウェアの設計と実装について述べる。第 8 章では、本システムによって指摘した問題点の解決が充足されるかどうか評価を行い、最後に第 9 章では本研究についてのまとめと、新たに発見された問題点と解決方針、また今後の課題について述べる。

Chapter 2

背景

2.1 インターネットメディアの変遷

インターネットインフラストラクチャの拡充につれて、コンテンツ視聴者が利用可能なネットワークメディアの種類が増加してきた。当初のインターネット接続方法は、UUCP 等による間欠接続が主であり、そこで通信される情報はテキスト情報のみであったが、次第に常時接続が主流になってきた。更にネットワークの帯域の高速化と、各種技術の整備により動画の配信も盛んに行われるようになってきた。

コンテンツを提供するメディアの種類が増加するに伴い、コンテンツ提供の方法も多様化してきた。当初は電子メールや電子ニュースによるテキストベースの通信が主たるコミュニケーションツールであったが、HTML による画像や文章レイアウトの実現、動画による動作の表現の様に表現の幅が広がってきた。

更に CU-SeeMe・MBONE 等 の出現によりストリーミング技術が注目を浴び、動画を逐次配信するという表現方法が加わった。

現在では、動画配信のためにストリーミング技術を利用する機会が増えて来ている。

2.1.1 ストリーミング技術の特徴

ストリーミング技術は、旧来の一括して転送するコンテンツ提供形態と異なり、必要な部分を逐次送信・逐次再生することを特徴とする技術である。

逐次送信・逐次再生の技術により、以下の利点が得られる [3]。

- 逐次送信の技術により、全てのコンテンツを受け取るまで待たなくて良いので視聴開始までの待ち時間が短い
- 逐次再生の技術により、コンテンツを視聴する側に大きなストレージが必要無い
- 逐次送信・逐次再生を実現するため、利用帯域に応じた内容を送信するので、コンテンツ全体を一度に転送しない。このため、コンテンツ全体をコピーされる可能性が低い。

このような利点を生かして多くの場面でストリーミング技術が利用されるようになった。

RealMedia における例 ここで、RealServer がどのようにコンテンツを送信するか確認する目的で実験した際のログを元に RealServer のファイルアクセスパターンを表したものを、「Figure 2.1 RealServer によるコンテンツファイルアクセス順」に示す。この実験の際には、およそ 64kbps の帯域でデータを取得した場合、最初にコンテンツファイル先頭の 65536bytes を取得した。その後は、ファイル中途から 65536bytes づつの取得を行なう系統の取得動作が確認できた。これらはそれぞれおよそ 20 秒間隔と 50 秒間隔でのファイル取得要求をしている。

この結果より、RealMedia の場合はコンテンツファイル内に複数の帯域のコンテンツが保管されており、視聴者の帯域によってどの部分を再生するかが選択されていると推測できる。

この推測が正しいと仮定すると、少なくとも RealMedia においてはコンテンツを開始から終了まで閲覧したとしても、コンテンツファイル内にアクセスされない領域があると言える。

2.1.2 ストリーミング技術の需要

ストリーミング技術は、インターネットラジオやインターネットライブ中継のような既存の放送メディアの放送手段をインターネット上で実現する手段として広く用いられるようになって来た。これは、提供者側から見て既存の放送メディアよりも新規参入コストが低いことや、One-on-One マーケティングが行えること等が理由である [16]。

加えてビデオオンデマンドと呼ばれるようなコンテンツ提供スタイルが定着してきている。現在では、遠隔教育環境において過去の講義を保管・閲覧

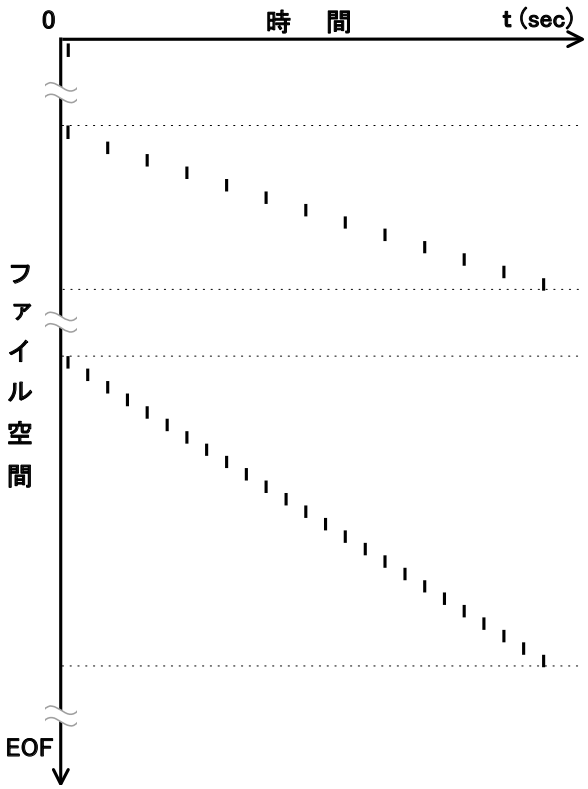

Figure 2.1: RealServer によるコンテンツファイルアクセス順

する手段などで利用されており、将来的には例えばレンタルビデオの代替手段として利用されることが研究されている [16]。また、パーキャス TV[14] のような、ミニ放送局を多数集めて運用するような試みも提案されている。このように今後も多様な形態が提案されることが予想される。

2.2 コンテンツホルダーとシンジケーションの分離

一つのシンジケーションが多数のコンテンツを扱う環境においては、そのシンジケーションが著作頒布権を保持するコンテンツ以外に、他のコンテンツホルダーが著作頒布権を保持するコンテンツを提供する場面が出てくる。

例えば遠隔教育環境において、他大学の講義を借用し自校の受講単位として利用する場合や、他大学と合同で講義を行うような場面が例として挙げられる。また、現状のマスコミュニケーションにおいて、テレビ局が番組製作会社のコンテンツを放送しているようなモデルも、シンジケーションがコンテンツホルダーではない例の一つである。

このように、コンテンツ提供者と配布者が異なる理由は、以下のように推測される。

提供インフラの問題 コンテンツホルダーが主にコンテンツ作成に従事する場合のように、提供するためのインフラを維持することが主たる業務でない場合が考えられる。この場合、コンテンツホルダーはインフラの構築に掛ける労力を避けたいと考えるため、インフラの整備されたシンジケーションに対して配布を依頼する。

視聴者の範囲を限定する場合 例えばイントラネット上での社内教育利用のように、視聴者が限定された環境でコンテンツを提供する場合が考えられる。この場合、セキュアな社内 LAN で使用する場合の様に、コンテンツを発信する場所がインターネットから到達不能な場所である場合も少なくない。

コンテンツを相互利用する場合 例えば遠隔教育環境のように、複数の組織で相互に講義内容を補完したい場合や合同で講義を行う場合が考えられる。この時、講義の視聴をもって単位として認める組織が視聴記録を取得したいと考えるため、実際のコンテンツ配布は単位を認める側で行えるようにすることが望まれる。すなわち、単位を認める側はシンジケー

ションであり、講義を行った側であるコンテンツホルダーとは組織が異なる。

以上から、コンテンツホルダーとシンジケーションが独立した組織で運用される場面が出て来たことが伺える。

2.3 コンテンツ提供ポリシの設定

コンテンツホルダーは様々な理由により、コンテンツに対して提供ポリシを設定する。例えば以下のようなものが考えられる。

- 対価として代金を徴収する目的で視聴者認証を行う場合
- コンテンツの理解度を深めるために視聴に対して前提条件を付ける場合
- マーケティングの材料として視聴者の属性情報が欲しい場合
- 有害情報対策として視聴者制限を行う場合
- イベントとしてコンテンツ提供する際に抽選で視聴コンテンツを振り分ける場合
- 著作頒布権保有者としてコンテンツの拡散を抑制する目的
- その他

これらを実現するために、コンテンツホルダーは視聴者認証や、視聴ログの保管、アクセスの監視を行う。

しかし、コンテンツホルダーとシンジケーションが分離することで、コンテンツのアクセスコントロールの管理は難しくなる。これは、コンテンツの分散保管を行うことで、コンテンツサーバが複数存在することとなり、アクセスの監視が行いづらくなることが原因である。また、シンジケーションが別組織であるため、現状ではコンテンツ借用契約の締結等により、相手の運用を信用することによってコンテンツの提供ポリシが実現されているにすぎない。

2.4 現状の問題点

ここで、コンテンツホルダーとシンジケーションが独立の組織で運用される場合において問題となる点を挙げる。

2.4.1　分散してしまう視聴ログ

単純にコンテンツをミラーリングした環境でコンテンツサービスを行っている場合において、その視聴ログは各コンテンツサーバの処理により、各ホストに蓄えられる。即ち、同じコンテンツを複数のホストよりサービスする場合、その視聴ログはその複数のホストに分散してしまうことになる。

このような状態では、あるコンテンツの視聴者をリストするために、その全てのミラーホストのデータを集積し、視聴ログを再構成する必要がある。これは結果の即時性が保証されず、またその収集のために相応の作業量が必要となる。

2.4.2　コンテンツ提供ポリシ表現の柔軟性欠如

コンテンツホルダーとシンジケーションが独立した組織で運用されている場合には、コンテンツホルダーが視聴ログを直接取得できない。このため、あるコンテンツの時間当たりの視聴回数を制限するようなポリシ表現や、ある特定のコンテンツを視聴することを条件に別のコンテンツを視聴可にするような、動的に視聴条件を変更するようなコンテンツ提供ポリシ表現を行うことができない。

2.4.3　コンテンツ拡散の危険性

ミラーリングを用いたコンテンツの分散環境においては、コンテンツホルダーがシンジケーションの運用を信用できない場合、シンジケーションの契約違反や管理の甘さによってコンテンツが拡散してしまう可能性がある。

このような環境において、視聴者の認証は各シンジケーションに任されるが、その認証設定が甘かった場合に、コンテンツホルダーが望まぬ相手からコンテンツを視聴されてしまうことがある。

また、シンジケーションが、コンテンツホルダーとの再配布禁止契約にもかかわらず、故意もしくは過失によってコンテンツを他のシンジケーションに対してコンテンツのコピーを許してしまった場合には、コンテンツが意図せぬ形で拡散してしまう。

このように、コンテンツ分散提供環境を実現する手法としてミラーリングを用いた場合には、コンテンツが拡散してしまう危険性が付きまとう。

2.4.4　認証運用のネットワークメディア依存性

現在の、コンテンツ視聴可否判断を行う機構は、コンテンツサーバに付属の機能を利用する場合がほとんどである。例えば、RealServer の認証機構を利用してユーザー名とパスワードを取得し、事前に登録してある暗号化済パスワードと照合し、適合するようならばコンテンツを配信するというような形である。

しかし、一般的にこのような認証機構は他の種類のストリーミングメディアを利用する場合に同じ認証機構を使用することはできない。例えばストリーミングメディアを変更する場合にはコンテンツサーバを変更する必要が出て来てしまうため、コンテンツサーバに付属の認証機構構造も変わってしまうためである。

同様に、例えば視聴者の環境に応じて高帯域用の技術と低帯域用の技術を選択する場合には、それぞれの技術を実現するために複数のコンテンツサーバを運用する必要がある。このような運用方法においては、全てのコンテンツサーバに付属の認証機構を利用する形で視聴者の認証を行う。しかしこの場合、コンテンツ提供ポリシを設定する際に各コンテンツサーバに応じた設定を行う必要が出てきてしまう。

このように、現在の認証機構の運用形態ではストリーミングメディアの移行や複数メディアを使用する際に問題があると言える。

2.4.5　現状の問題点のまとめ

本研究の背景として、現在ストリーミング技術が普及しつつある段階にあり、コンテンツホルダーとシンジケーションが分離する傾向が見られること、それによって以下の問題点が確認できることを挙げた。

- コンテンツ提供ポリシ表現の柔軟性欠如
- 分散してしまう視聴ログ
- コンテンツ拡散の危険性
- 認証運用のストリーミングメディア依存性

Chapter 3

関連研究

ここでは、ストリーミングコンテンツを複数組織で提供する場合に使用する技術と、コンテンツ提供ポリシを設定するときに使用する技術について個別に議論する。最後に各技術を比較し、「2.4 現状の問題点」で挙げた問題の解決に貢献するかどうかを検討する。

3.1 複数組織で提供するもの

まず、シンジケーションとコンテンツホルダーが異なる環境でコンテンツを配信するモデルの関連研究について説明する。

3.1.1 mirror

コンテンツファイルを単純に複数ホストにコピーし、それを提供する形を考える。この場合に使用されるのは多くの場合、mirror と呼ばれるスクリプトであり、これはコピー元とコピー先の双方のファイルツリーを比較し、差分を転送するという動作を行う。

この方法ではコンテンツ提供ポリシの設定はできない。CERN-httpd や Apache 等で用いられる .htaccess のように、コンテンツディレクトリ毎にコンテンツ提供ポリシを記載したファイルを保管し、提供時にそのファイルを参照することでポリシを実現するような仕組みを用いた場合にはその限りではない。しかし、この場合は静的なコンテンツ提供ポリシの実現は可能であるものの、「2.4.2 コンテンツ提供ポリシ表現の柔軟性欠如」で述べた動的な視聴条件変更は難しい。

また、「2.4.1 分散してしまう視聴ログ」で示したログの分散は避けられない。

更に、コンテンツとして提供するべき全てのコンテンツを各ホストに保持する必要があり、ここにかかる記憶領域の消費も負担となる。

3.1.2 分散ファイルシステム

コンテンツファイルを提供するのに分散ファイルシステムを用いる場合を考える。ここで言う分散ファイルシステムとは、Sun NFS(Sun Microsystems Network File System) や AFS(Andrew File System) などの、ネットワーク上で分散した箇所にあるファイルリソースを透過的に扱うためのファイルシステム一般を指す。

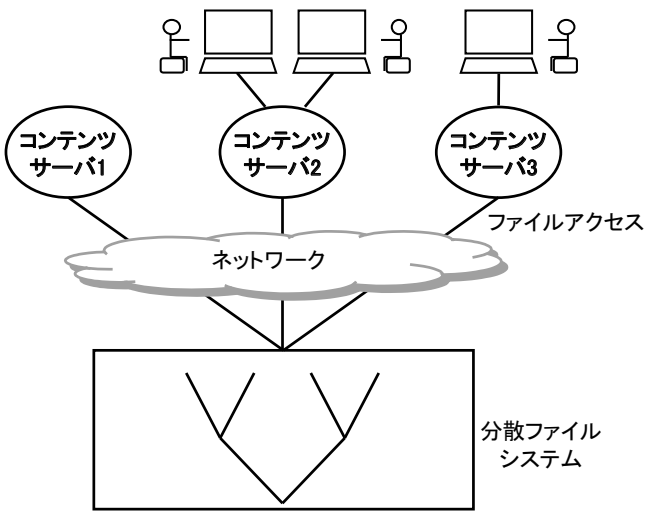

Figure 3.1: 分散ファイルシステム

分散ファイルシステムを用いて、コンテンツ配信を行う場合のモデルを「Figure 3.1 分散ファイルシステム」に示す。

各コンテンツサーバが動作するホストにおいて、コンテンツサーバがアクセスするファイルツリーを分散ファイルシステムを利用したものにする。そのファイルツリーより、コンテンツホルダーが提供するコンテンツファイルにアクセスできるようにする。これによって、コンテンツサーバがメディアクライアントからの視聴要求を受け付けた際にコンテンツホルダーよりコン

テンツの提供を受けることができるようになる。

この方法は、コンテンツファイルはコンテンツホルダーから提供を受けるため、一義的にはコンテンツの分散は避けられるが、通常のファイルシステムとして利用されているため、ファイルのコピーは比較的簡単である。また、「3.1.1 mirror」で述べた方法と同様に、ログの分散は避けられず、動的な視聴条件の変更も難しい。

3.1.3 RealProxy

RealProxy [8] は RealMedia 形式エンコードされたストリーミングコンテンツを中継するために用いられる機構である。これは主に、帯域を節約するためやプライベートネットワークにコンテンツを再送するために用いられる。

RealProxy は内部にコンテンツキャッシュを持ち、同じコンテンツに対する視聴要求があった場合にはキャッシュにあるコンテンツを再送する。ただし、コンテンツホルダーの意向を尊重するため、キャッシュから再送する場合であっても、メディアクライアントに対する配信の前に、必ずサーバーに対し視聴許可の伺いをすることで、コンテンツ提供ポリシの実現を行っている。

RealProxy は RealMedia の形式のみに対応し、その他のストリーミングメディアには対応していない。

3.2 柔軟なコンテンツ提供ポリシを設定できるもの

ここでは、コンテンツホルダーが動的な環境変化に合わせてアクセスコントロールを行う機能を持つ関連研究について説明する。

3.2.1 Apache において RDBMS を利用するアクセスコントロール

これは、ストリーミング技術で利用するものではないが、参考として Apache httpd での認証に RDBMS(Relational DataBase Management System) での解決を加えたものを挙げておく。これは、Apache の拡張モジュールとして、PostgreSQL を用いる AuthPG [13] や、Oracle を用いる mod_auth_oracle などを使用することで、ウェブページの視聴者をユーザー名とパスワードの対の正否を確認するための仕組みである。

通常の httpd 認証では、ユーザ名とパスワードの対が合致しているか否かのみを判定して、ウェブページ閲覧の可否を判断していた。しかし上記仕

組みを通じて、パスワード検証に RDBMS を用いることで、より柔軟な提供ポリシを記述することができるようになった。この仕組みにおいては、ユーザー名とパスワードが合致するタプルが存在するか否かをもって正否を判断する。

しかし、これらモジュールはテーブル検索時に発行するデータベース検索文の条件節を記述することでテーブル結合を結合をおこない、更に複雑なポリシ記述ができるようになる。

これを用いて、例えば視聴時間帯の制限や視聴回数の制限・あるウェブページを閲覧済であることを条件に閲覧を許可するというような、動的なアクセスコントロールをすることができる。

3.2.2 RealServer 認証機能

RealServer 認証機能 [7] は通常の RealServer に、ユーザ認証機能とデータベースの接続インタフェースを備えたものである。これを利用することで、RealMedia 形式にエンコードされたコンテンツに対して「3.2.1 Apache において RDBMS を利用するアクセスコントロール」で述べたような動的アクセスコントロールができるようになる。

RealPlayer には各クライアントに対して一意な Client_ID が振られているため、通常のユーザー名とパスワードの対を検証する方式に加え、視聴者認証を行う場合にそれを利用した視聴者特定方式を利用することもできるようになっている。これによって、視聴者が操作せずとも視聴者を特定してアクセスコントロールをできるようなコンテンツ提供ポリシを設定することができる。

ただし、この仕組みは RealMedia 形式に特化されているため、他のメディア技術を使用する際にはこの認証機構は使用できない。

3.2.3 Kerberos

Kerberos[9] は対称鍵暗号系を用いた principal [1]同士が相互に認証を行うための仕組みである。principal は他の principal を認証したい時に、KDC(Key Distribution Center 鍵配布センタ) に対して、認証の仲介を依頼する。KDC は各 principal の鍵を管理しており、認証の仲介依頼に応じて各鍵を使用し、認証に必要な情報を提供する。

その仕組みの概要を「Figure 3.2 Kerberos 概要」に挙げる。

[1]Kerberos の利用者と資源の総称

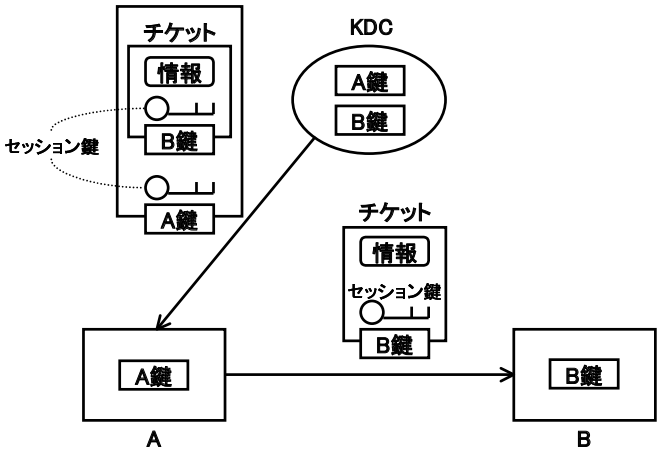

Figure 3.2: Kerberos 概要

具体的には、認証依頼を受けた場合に、セッション鍵と呼ばれるその認証プロセスのみで使用する鍵と、チケットという認証先で使用する情報を組み合わせたものを依頼者の鍵で暗号化し、依頼者に返却する。依頼者は上記情報のうち、チケット部分を認証対象者に送付する。チケットはセッション鍵とその認証の有効期限等の属性を認証対象者の鍵で暗号化したものであり、認証対象者がセッション鍵を得ることができることで、KDC に保管されている認証対象者の鍵と認証対象者が保持している鍵が一致することを確認できる。すなわち、認証依頼を行った者がこのセッション鍵を使った通信を行えることで、認証対象者が正しく KDC に認められた principal であることが保証できる。

Kerberos を使用するためには、Kerberos の提供するインタフェースに対応したアプリケーションを作成する必要がある。

3.2.4 WMRM(Windows Media Rights Manager)

WMRM[15] は、マイクロソフトが提案する著作権処理システムの為のソフトウェア群であり、SQL を用いたデータベースを用いて、複雑なコンテンツ提供ポリシの記述を行うことができる。WMRM においてコンテンツは、鍵を使用し暗号化された形で配信される。コンテンツを再生するためには、

WMRM に対応したメディアクライアントが必要である。最初の再生時に、コンテンツホルダーの認証サーバと通信を行い、設定された使用許可条件を送付しておく。メディアクライアントはその使用許可条件に適合する場合には再生を行う。

WMRM はストリーミングに特化した仕組みというわけではなく、コンテンツを一通り取得し、ローカルファイルとして保持しているものを再生する場合に適用することを前提とした仕組みになっている。このため、暗号化されたコンテンツそのものは拡散することを認めており、またコンテンツの視聴毎にコンテンツサーバに接続をするわけではないので、構造的に視聴ログは残らない。

また、メディアクライアントはこれに対応したものを使用する必要があり、現状では Windows 上でのみ、これを開発できる環境が整えられているため、他の OS 上で WMRM を使用した暗号化コンテンツを再生するのは難しい。

なお同様に、暗号化したコンテンツを専用のメディアクライアントで再生するというアプローチを採用している技術として、以下のようなものもある。

- **liquid audio** [2]
- **MagicGate** [3]

3.3 関連研究の比較

ここで、要求に対する関連研究の比較を行う。

「Table 3.1 関連研究の比較」に、ストリーミングコンテンツを提供する際に各技術を利用した場合に、前述の問題点がどの程度充足されるかを表す。

なおここでは、ストリーミングコンテンツの配信を行うモデルではない「3.2.1 Apache において RDBMS を利用するアクセスコントロール」と「3.2.3 Kerberos」については除外してある。また、「3.2.2 RealServer 認証機能」と「3.1.3 RealProxy」についてはこれらを組み合わせて使用する環境を想定する。

以上から分かる通り、本研究において注目すべき、コンテンツ提供ポリシの柔軟性とメディアの非依存性を兼ね備えたストリーミングコンテンツサーバは発見できなかった。

[2] http://www.liquidaudio.com/
[3] http://www.world.sony.com/JP/News/Press/199902/99-028/index.html

	ポリシ柔軟性	ログ集中	コンテンツ拡散防止	メディア非依存
mirror	× ：ポリシ反映はできないことはないが高度なポリシ設定はできない	× ：ログは分散して保管される	× ：根源的にコンテンツが拡散している	○ ：どのようなメディアにも対応できる
分散ファイルシステム	× ：mirror と同様	△ ：ログファイルを共有すればログを集中させられないことはない	△ ：オリジナルコンテンツ自体は拡散していないが、ファイル操作によるコピーは容易	○ ：mirror と同様
RealServer 認証 + RealProxy	◎ ：関係データベースを用いた柔軟なポリシ設定が可能	◎ :RealProxy に対するアクセスログも元 の RealServer が動作しているサイトに保管される	○ ：途中通過形式はストリーミング形式であるため、コピーは難しい	× ：RealMedia のみに対応
WMRM	◎ ：関係データベースを用いた柔軟なポリシ設定が可能	△ ：Manager の設定によりログを集中することは可能	△ ：ストリーミングによる提供であるためコピーは難しいが、独立した別組織における頒布は考慮されていない	△ ： WMRM に対応するよう作成されたコンテンツサーバが存在するメディアでは対応可能

Table 3.1: 関連研究の比較

Chapter 4

概要

4.1 研究の概要

本研究は、コンテンツホルダーがコンテンツを提供する場合、現在使用されている技術では著作頒布権を正しく行使するための構造が不十分であり、またコンテンツ視聴に制限を掛ける場合の条件付けが柔軟でないことを問題としている。特にコンテンツホルダーとシンジケーションが別の組織である場合に、現状ではコンテンツホルダーはシンジケーションのコンテンツ配信の運用を信用する必要がある。このような環境下では、コンテンツがコンテンツホルダーの意図以上に拡散してしまう可能性が高まってしまう。

そこで、コンテンツホルダーによるコンテンツ提供ポリシの記述を行う仕組みを実現する方法について調査を行った結果、現状ではコンテンツ提供ポリシはメディア毎に異なる記述法をとる必要があり、また動的なコンテンツ提供ポリシの実現を行うことができるメディアは限られていることが分かった。

このような制限を取り除くために、汎用的に動的なコンテンツ提供ポリシを実現するモデルについて提案する。またこの提案の正当性を証明するため、当該モデルを元にシステムを設計・実装し、既存のシステムとの比較評価を行った。

このモデルを利用することで、過去の視聴情報を保管し、それを元にポリシ設定を行うことで、動的な視聴条件を設定することができるようになる。

4.2　本研究の特徴と期待される成果

シンジケーションとコンテンツホルダーが独立した組織である環境で、ストリーミングコンテンツを提供するための構造は、大きく 2 つに分類できる。それは、コンテンツファイルそのものを分散保管させるものとコンテンツサーバそのものにコンテンツ提供ポリシを解釈する構造を持つものである。前者は柔軟なコンテンツ提供ポリシを設定できず、更にコンテンツの不正なコピーが容易であるためコンテンツが拡散しやすいという欠点を持つ。後者は提供ポリシの設定対象がメディアに依存しまうという問題がある。

対して本研究は、メディアに依存せず柔軟なコンテンツ視聴ポリシを設定できる環境を提供することが期待できる。

本研究では、ここで検討したモデルを実際に設計・実装することによって、このモデルの正当性を検証する。

Chapter 5

構成

ここで、前述「2.4 現状の問題点」を考える上で、必要な各要素について検討する。

5.1 構成要素

ここで、本研究を考察する上で必要な構成要素について整理する。
「Figure 5.1 構成要素関係図」に表す関係に沿って説明する。

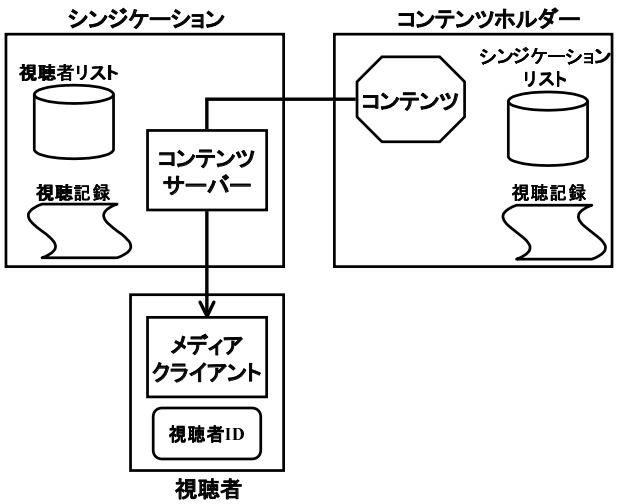

Figure 5.1: 構成要素関係図

コンテンツホルダー コンテンツホルダーはコンテンツ作成者もしくはその者が所属する組織、または著作頒布権を委託された者である。コンテンツホルダーは、利益を得る目的のため、もしくは様々な不利益を被らないようにするために、コンテンツの提供に制限を掛ける。

シンジケーション シンジケーションは視聴者に対して十分なコンテンツを提供し、それをもって直接的もしくは間接的に利益を得ることを目的とする。シンジケーションはコンテンツを配布することを目的とした組織であるため、コンテンツの作成は行わない、もしくは一部を行うことを前提とする。このため、コンテンツホルダーより全部もしくは一部のコンテンツを借用する必要がある。

視聴者 視聴者は、必要な知識を手に入れること、もしくは楽しみを得ること等を目的としてコンテンツを視聴する。視聴者が有価値なコンテンツを視聴するためには、いずれかのシンジケーションより認められることによって、視聴する権利を得る必要がある。

コンテンツ コンテンツはコンテンツホルダーよりシンジケーションに対して、視聴者に見せることを目的に貸与される。一般に、コンテンツそのものが無制限に拡散されることは望まれない。

コンテンツホルダーの視聴記録 例えばコンテンツ配信ビジネスを行う際に、あるシンジケーションに対し提供総数や時間あたり提供数を制限する場合がある。これらを考慮に入れ、コンテンツホルダーは視聴記録を保持しておくことができるようにする必要がある。

シンジケーションの視聴記録 例えば、遠隔教育環境において単位認定を行うためには、単位を発行する主体であるシンジケーションが、正しく学生の視聴記録を確認できる必要がある。これを実現するため、シンジケーションにおいても視聴記録を保管しておく必要がある。

コンテンツサーバ シンジケーションにおいて、ストリーミングコンテンツを提供する役割を持つコンテンツサーバを動作させる。視聴者はこのコンテンツサーバと通信することで、目的のコンテンツのサービスを受けることとなる。

シンジケーションのリスト コンテンツホルダーは、接続されて来たシンジケーションが信用できる者かどうかを確認するためにシンジケーション

のリストを持っておく。シンジケーションからのコンテンツ提供要求がある際には各属性がこのリストと符合するかどうか確認して、適正であると判断できた場合に提供を行う。

視聴者のリスト シンジケーションは視聴者が自らの管理下にある者であるかどうかを確認するために視聴者のリストを持っておく。視聴者がコンテンツを視聴する際には、シンジケーションが視聴者情報をこのリストと照合して適正であると判断できた場合に提供を行う。

メディアクライアント 視聴者は、コンテンツサーバから送られてくるコンテンツをメディアクライアントを通じて視聴する。

視聴者 ID シンジケーションやコンテンツホルダーに対して、視聴者が誰であるかを知らせるために必要となる識別子である。

5.2 コンテンツ提供ポリシの種類

「2.3 コンテンツ提供ポリシの設定」で述べたように、コンテンツホルダーは様々な理由でコンテンツ提供ポリシを設定することを望む。それは例えばコンテンツホルダーの利益確保のためや、コンテンツに付加価値を与えるため、視聴者の属性をマーケティング材料とすることでコンテンツの質的向上に努めるためである。

このような要求を満たすシステムを構成する上で、現実社会での要求とそれを実現するための条件の整理が必須となる。

ここで現実社会でコンテンツ提供ポリシを設定する場合に可能性のある形態を例示する。また、各ポリシ設定を実現するためにどのような情報を必要とするかを挙げる。

5.2.1 コンテンツホルダーとシンジケーション間

ここでコンテンツホルダーとシンジケーションとの間で設定され得るポリシ設定を挙げる。

配布代行

コンテンツホルダーが、視聴者情報の管理代行依頼や帯域幅の不足等の問題を理由に、シンジケーション対してコンテンツ配布の代行を依頼する場合が考えられる。これは、コンテンツホルダーが衆目を集めていないために、

シンジケーションの視聴者の層の厚さを期待し、コンテンツの配布をコンテンツホルダー名の下に行う場合も含めて考えられる。この場合においては、配布した相手が誰であるか、どの程度の回数の配布が行なわれたかなどの結果をコンテンツホルダーが信頼できる形で確認できるようにすることが望まれている。

配布権販売

コンテンツの配布権を、配布数や配布期間に応じた形などで与える形式が考えられる。これは、主に企業組織内で使用されることを目的としたイントラネットのような環境で、内部の人間を対象にコンテンツを配布することに注目した場合に取られると考えられる契約形態である。配布権を販売する場合には、そのコンテンツの配布回数・期間が契約と適合しているか、視聴者はコンテンツホルダーが想定した範囲内の人間であるかどうかをコンテンツホルダーが信頼できる形で確認することができるようにすることが望まれている。

相互乗り入れ

例えば遠隔教育環境において単位互換制度を導入する場合のように、あるコンテンツを自らのコンテンツの様に扱いたい場合が考えられる。これは、シンジケーションにおいてもコンテンツ作成業務を行っているが、その作成範囲がシンジケーションの対象とする人間にとって不足である場合が考えられる。

権限委譲

あるコンテンツに対して配布に関する全ての権限を委譲し、以後の配布・管理等は全てシンジケーションに任せる場合が考えられる。この場合は、コンテンツホルダーは権限委譲後は権利を主張しないことが前提となるため、権限委譲の可能性に対応した仕組みを構築する必要はない。

5.2.2 シンジケーションと受講者間

身分保証

コンテンツホルダーに対して、正しくシンジケーションの管轄下にある人間であることを保証する場合が考えられる。例えば遠隔教育環境において

は、ある学生が所属教育機関の学生であることを保証する場合がこれに相当する。正しく視聴者がシンジケーションの管轄下にあることを保証することによって、シンジケーションはコンテンツホルダーに対して、契約に基づいて視聴者に正しくコンテンツ提供をしていることを保証することができる。

包括視聴契約

シンジケーションが、契約上提供できる全てのコンテンツを受講者に対して提供する契約形態が考えられる。この場合は、視聴者が正しくシンジケーションにより認証されている事実がコンテンツホルダーに対して証明できれば、コンテンツホルダーはコンテンツを安心して提供することができる。

各個視聴契約

シンジケーションが契約上提供できるコンテンツに対して、各々視聴の可否を行う場合が考えられる。ここでは、コンテンツホルダーはシンジケーションに対しては視聴許可を与えていることになるので、シンジケーションがその視聴を許可すれば視聴者に対してコンテンツを提供しても良いことになる。

5.2.3 コンテンツホルダーと受講者間

包括視聴契約

コンテンツホルダーとシンジケーション間の契約の如何に関わらず、視聴者に対して全てのコンテンツを提供する場合が考えられる。ここでは、当該視聴者がコンテンツホルダーにとって正しく契約対象者であることが確認できれば、コンテンツの提供をしても良いことになる。

各個視聴契約

コンテンツホルダーとシンジケーション間の契約の如何に関わらず、視聴者に対して各個のコンテンツを提供する場合が考えられる。

5.3　コンテンツ提供ポリシ設定

5.3.1　ポリシ設定の種類

ここで、コンテンツホルダーが設定する提供ポリシの種類を挙げ、各々の設定をする際に必要となる情報について整理する。

正確な視聴記録が必要なもの　コンテンツ視聴の対価として料金を徴収することが目的のものや、コンテンツ視聴を条件に具体的な利便を与えることが目的のものがこれに相当する。具体的な利便とは、例えば教育環境における受講単位の付与や、対価として特典を与える場合などが考えられる。

これらは、正しい視聴者認証に基づいて視聴記録を保管し、それに基づいて対価を請求もしくは付与することができるようにしなくてはならない。また、視聴権限が無い者に対するアクセスの拒否は厳密に行う必要がある。この場合、コンテンツホルダーがコンテンツ提供時に要求する情報は以下の通りである。

- 正確な視聴者認証情報
- 視聴時間に関する記録

視聴者傾向を得るもの　あるコンテンツに対してどのような者が興味を持つか、あるいはマーケティング材料として視聴者の層や各属性と視聴コンテンツとの相関関係を知りたい場合などがこれにあたる。マーケティングの材料として属性が欲しい場合には、各個に厳密な視聴者認証は必要無く、属性は統計情報として得られれば良い。この場合、コンテンツホルダーが要求する条件は以下の通りである。

- 視聴者属性
- 視聴開始時間や終了時間などの視聴行動パターン
- コンテンツ視聴情報の統計

視聴者属性によって視聴制限を掛けるもの　これは例えば、視聴者が未成年である場合に有害コンテンツを見られないように制限を掛けるものや、視聴者が特定の外国居住者である場合、宗教や政治上の問題によって視聴制限を掛けるような場合が考えられる。この場合に要求される情報は以下の通りである。

- 視聴者属性

コンテンツホルダーの意図によって視聴制限を掛けるもの 視聴時間帯制限や同時提供数制限など、主にコンテンツホルダーのサーバ状態によって視聴制限を掛けるものが考えられる。これは、サーバリソースの保全を意図したものの他、視聴ライセンスの提供形態によるものが考えられる。この場合、アクセスコントロールをする際に要求される情報は以下の通りである。

- コンテンツホルダーサーバの状態情報
- 視聴者属性

諸条件によって視聴コンテンツを振り分けるもの 例えば、くじのイベントなどで「あたり」と「はずれ」のコンテンツを提供するためにコンテンツを入れ換えることを考える。この際に、コンテンツサーバは同じ内容のアクセスで、提供するコンテンツを変更することで対応する。この場合には、他視聴者の過去の結果記録や当該視聴者の前回結果などを加味して提供内容を変更する必要がある。

また、視聴可否判断の結果として視聴拒否等がされた場合に、本来のコンテンツと異なるものを出力することで、異常があったことを知らせることができるように考える。この場合は、加えて異常の状況を示す情報となる、エラーコードをシンジケーションに知らせる必要がある。

- 視聴者情報
- 過去の視聴記録
- 異常の状況情報

5.3.2 ポリシ設定に必要な情報

以上で見てきたように、コンテンツ提供ポリシ設定に必要な情報は以下のようなものがある。

- 正確な視聴者認証情報
- 視聴時間に関する記録
- コンテンツ視聴情報の統計
- 過去の視聴記録

- 視聴者属性
- 視聴開始時間や終了時間などの視聴行動パターン
- コンテンツホルダーサーバの状態情報
- 視聴者情報
- 異常の状況情報

これらの情報は、「5.1 構成要素」で述べた関係性に従って整理すると、以下に挙げる情報が取得できれば、その情報から抽出できることがわかる。

- 視聴者情報と属性 (視聴者リストによる)
- コンテンツ視聴記録 (視聴記録による)
- サーバの状態情報
- エラーコード

本研究では、上に挙げたこれらの情報をコンテンツホルダーが取得できる構造についても検討する。

Chapter 6

モデルの設計

ここでは、「2.4 現状の問題点」に挙げた各問題を解決するためのモデルを設計する。

6.1 モデル設計の前提

ここで、本システムを設計する上で、前提条件となる事柄に付いてその理由とともに説明する。

6.1.1 既存ソフトウェアの使用

ストリーミングコンテンツを扱うためのプロトコル・ソフトウェアは多様な種類のものが提供されている。それぞれに一長一短があり、全ての状況において一律にこれを使えば確実に最高の性能が得られるというソフトウェアは存在しない。

ストリーミングに要求される性能は、使用できる帯域に依って全く異なる性質のものとなる。

例えば、低速域ではコンテンツの内容を理解させるために最低限の情報である音声に比較的重点を置いた技術を用いる傾向がある。しかし、高速域においては、ある程度以上の品質が確保されていれば、音質よりも画質の向上に重点をおいた技術を用いる方がよい。これは、コンテンツをより良く認識するために必要な一定の品質を確保することを考える場合、音質を一定水準まで確保するために必要な情報量と、画質を一定水準まで確保するために必要な情報量とでは明らかに後者の方が大きい情報量が必要であるためである。即ち、既に一定以上の品質を確保されている音質を更に向上させること

を考えるより、より低い品質で通信されていると考えられる画質の確保を考える方が良いということとなる。

以上から、使用できる帯域に応じた技術を採用する必要がある。更に、インフラストラクチャは今後とも高速化・多様化していくことが予想されるため、今後は現在とはまた違った方向性の技術が開発・普及することが予想される。

ここで視聴者情報を認証するシステムを新たに導入するために、ある技術に特化した構造を採用すると、その他の帯域を主眼に置いたコンテンツに対応することができなくなると考えられる。このため、本研究では特定のストリーミングコンテンツ配信技術に依存しない設計をする必要がある。

また、ストリーミング技術のように、ある程度以上の技術的複雑さと使用者の数が見込まれる場合には、その技術を販売することを目的として開発されることも充分に考えられる。この場合には、使用許諾条件等に依りソフトウェアの改編を認めない場合やプロトコルの解析を認めない場合がほとんどである。このようにストリーミング技術そのものに販売されているものを使用する場合にはシステム上の問題に係わらずソフトウェアの変更は不可能となる。

即ち、上記より既存のコンテンツサーバ・メディアクライアントの構成をそのまま利用できるモデルを考案する必要がある。

6.1.2 URI 指定に Web システムを使用する

本研究では、説明の簡単のため、ストリーミングコンテンツそのものの検索指定を行う際に Web システムを使用することを想定する。

視聴者は Web ブラウザを通して視聴できるコンテンツの候補を取得し、その中から視聴するコンテンツの URI(Uniform Resource Identifier) を選択する。URI はコンテンツの種類や保管先、その識別情報を含んでいる。Web ブラウザは URI に記述されている情報から、どのメディアクライアントを動作させるべきかを判断する。メディアクライアントは URI を受け取ることで、どのコンテンツを再生するべきかを判断する。これにより、シンジケーションはメディアクライアントに対してコンテンツの情報を知らせることができるようになる。

しかし、上記における Web システムと同様な機能を持つものあれば、コンテンツ候補情報をメディアクライアントに対して送る機構として別のものを使用することができる。

6.2 要求上の問題

6.2.1 必要な情報

コンテンツ提供ポリシを設定する上で要求される情報は、「5.3.2 ポリシ設定に必要な情報」で述べた通り、以下の通りである。

- 視聴者情報と属性 (視聴者リストによる)
- コンテンツ視聴記録 (視聴記録による)
- サーバの状態情報
- エラーコード

6.2.2 視聴者の本人性確認

視聴者が正しく本人であるかどうかを、コンテンツホルダーが納得できる形で示す必要がある。視聴者が本人であるかどうかを確認するためには、ID とパスワードの対の認証や証明書の有無等の情報を用いる。

このような視聴者情報をコンテンツホルダーが不可逆変換して他人には解釈不可能な形に直し、それを視聴者に渡す。次回視聴者が視聴する際に、視聴者情報とともに当該変換結果を提示する。コンテンツホルダーが、視聴者情報を暗号化したものが当該変換結果と一致することで、コンテンツホルダーは視聴者が本人であると確認できる。

しかし、「6.1 モデル設計の前提」に述べた制限である、既存ソフトウェアと枠組を使用するという条件により、視聴者の端末において積極的に情報の提供や暗号化・平文化等の作業はできない。

このため、シンジケーションもしくはコンテンツホルダーで代理にこのような作業を行う必要がある。

視聴者情報の渡し方

コンテンツサーバとメディアクライアントに既存のソフトウェアを使用すると考える場合、これらから視聴者情報を得ると想定することはできない。また、同様にこれらコンテンツ視聴者を特定するのに一意な ID があると仮定することはできない。このため、何がしかの方法により視聴者から視聴者情報を得ることを考えなくてはならない。

視聴者の端末において、コンテンツ閲覧時に確実に動作していると考えられるアプリケーションは Web ブラウザとメディアクライアントに限られると

考えて良い。そして、Web ブラウザとメディアクライアント間で受け渡される情報は URI のみである。このため、URI を用いて視聴者を識別しなくてはならない。

URI の要素中、他の情報を含めることができる可能性がある部分は、ファイル名部分に限られる。

このため、本研究の前提の中で視聴者情報を導入するためにはファイル名に情報を埋め込むという選択肢を取ることが最善策であると考えられる。

6.2.3 なりすましやコンテンツ再送の防止

視聴者を正しく認証するためには、当該視聴者へのなりすましや、通信路を流れる情報を盗聴してコンテンツ要求を再送する攻撃に対して堅牢である必要がある。このために、コンテンツを転送中にもコンテンツホルダーはシンジケーションの確認を行いつつ送信する必要がある。これを実現するためには、公開鍵暗号系を用いた相互認証が有効であると考えられる。

しかし、暗号化は計算機資源を多く使用する操作であるため、特にストリーミングコンテンツの送信のように多くの情報を通信する際には暗号化のコストは掛けないことが理想である。

上記に基づき、本研究においては各要求部分を信頼できる形で通信することを主眼に考える。

6.3 設計の方向性

本システムを実現するためには、コンテンツ要求に対して、アクセスコントロールを行うための視聴者認証処理と適切なコンテンツ提供ポリシの反映を行う必要がある。それに加えて「6.1 モデル設計の前提」において定義された要求項を正しく満たすためには、既存の構造を大幅に変更すること無く、既存構造の内、いずれかの場所に認証処理を行う機構を付加する必要がある。

ここで、上記機構を付加する位置について、コンテンツサーバ・メディアクライアント間に位置するポートフック案と、コンテンツサーバ・ファイルシステム間に位置するファイルシステム案の二つの可能性を挙げる。

6.3.1 ポートフックを利用した実装案

ポートフックとは、メディアクライアントからコンテンツサーバに対して通信を行う際に、コンテンツサーバの代理として一度別のシステムが着信を

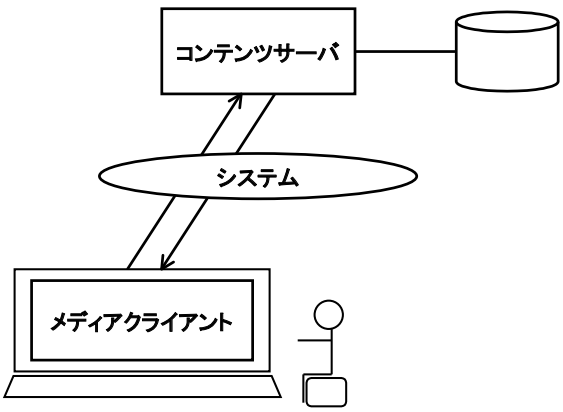

Figure 6.1: ポートフック案

行い、適切な処理を行った上でコンテンツサーバにその処理を継続する方法である。すなわち、「Figure 6.1 ポートフック案」で示したように、メディアクライアントとコンテンツサーバの間にシステムの導入を行う場合を意味する。ポートフックを用いることで、メディアクライアントからの通信要求を元に認証処理のきっかけを作ることができるようになる。

しかし、既存のソフトウェアの組合せを使用しながら、ポートフックを行うシステムに対し新たな情報を提供、もしくは付与するのは難しい。既存のソフトウェア間の通信プロトコルは既存ソフトウェア間において解釈可能なものが既に規定されているが、それに新たに視聴者の認証を行うための情報を付与するためには通信プロトコルを一部変更する必要がある。プロトコルを変更するためにはコンテンツサーバとメディアクライアントの変更が必要となって来るため、これは「6.1.1 既存ソフトウェアの使用」で示した前提に反する。

また、通信プロトコルはコンテンツメディアの種類に依存する。もし通信プロトコルを変更しそこに視聴者の認証情報を付け加えようとする場合には、同時に転送状態や速度などの情報が得られる反面、各コンテンツメディアで使用されているプロトコル各々について変更を加えなくてはならない。即ち、コンテンツメディアの数だけ本システムの派生型を作成する必要が出てきてしまう。また将来新たに作成されるであろうコンテンツメディアに対応でき

ない。このようにコンテンツメディアに依存する形の実装を行わなくてはならない状況は、汎用のシステムを作成する上で足枷となる。

6.3.2 ファイルシステムとしての実装案

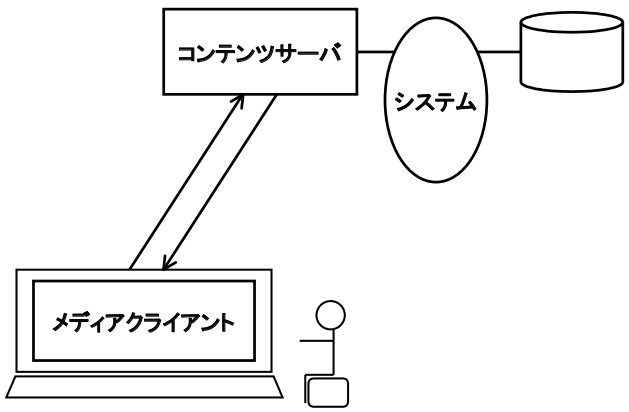

Figure 6.2: ファイルシステム案

他の方法として、ファイルシステムの一部に認証の仕組みを導入する方法が考えられる。すなわち、「Figure 6.2 ファイルシステム案」で示した、コンテンツサーバがファイルアクセスを行うことをきっかけとしてシステムが機能する場合を意味する。これは、コンテンツサーバがメディアクライアントからの要求に応じてコンテンツを提供する際に、コンテンツサーバがファイルシステムに対してファイル読み込みの要求を行うことに着目した案である。

視聴者がコンテンツサーバに対してサービスを要求した場合に、コンテンツサーバはコンテンツを取得するためにファイルシステムに対してファイルオープンの要求を行う。この時ファイル名の一部として視聴者の認証情報等を含めておく。その情報を基に視聴者の認証作業と、該当コンテンツの視聴可否の判断を行うという方法を考える。

構造上、この方法では「6.1.1 既存ソフトウェアの使用」で示した、コンテンツサーバとメディアクライアントの変更は行わないという条件を満たすことができる。

その反面、ファイル名の一部として視聴者の認証情報を含めるための仕組

みが必要になる。これはコンテンツを指定する為に訪れる Web ページ を動的に構成し、リンクとして張られるコンテンツ URI に予め認証情報を構成するために必要な情報を含めておくという方法によって解決できる。

6.3.3 本モデルでの設計

	ポートフック案	ファイルシステム案
通信プロトコルの変更	必要	不要
サーバ・クライアント変更	必要	不要
メディアに依存	する	しない
メディア依存情報の取得	できる	できない

Table 6.1: ポートフック案とファイルシステム案の比較

「Table 6.1 ポートフック案とファイルシステム案の比較」に示した通り、各候補には欠点と利点がある。

これを踏まえて、本研究においてはファイルシステムの一部として認証の仕組みを導入する。

これは、複数種類ある既存のコンテンツメディアの配信プロトコルに対し、変更無く対応できることを重視したためである。「6.3.1 ポートフックを利用した実装案」で述べたように、認証情報を付加するのにポートフックの方法を用いる場合には通信プロトコルの変更をする必要が出て来てしまう。これは、既存のコンテンツサーバ・メディアクライアントの変更を伴わなければ実現できないため、要求項目を満たせないこととなる。

対して、「6.3.2 ファイルシステムとしての実装案」で述べたように、ファイルアクセスをきっかけにして認証を行う場合には、既存のコンテンツサーバ・メディアクライアントの変更を伴わずに認証情報を付加できると考えられる。

6.4 モデルの構成

ここで、設計の概略を説明する。

本モデルでは、コンテンツホルダーのホストで動作するマスター認証サーバ、シンジケーション上で動作するファイルシステム、シンジケーション上で取得する URI を指定する動的ページ生成機構からなる。全体像を「Figure

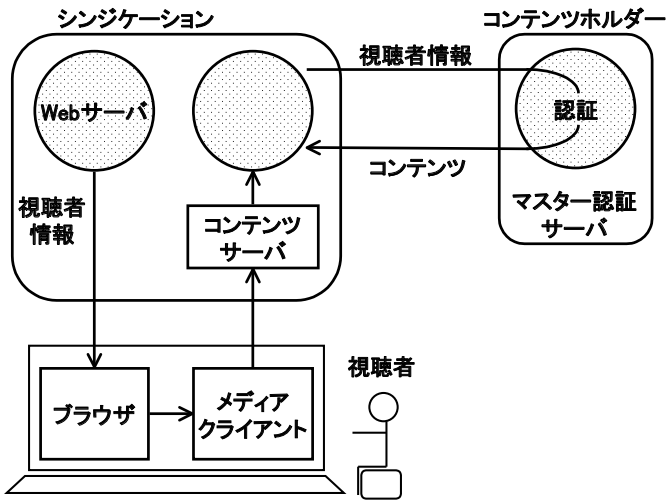

Figure 6.3: 本システム概要

6.3 本システム概要」に示す。

視聴者は Web ページを通じて目的のコンテンツを視聴する。視聴者がコンテンツを視聴しようとする際に、目的のコンテンツを表す Web ページ上の URI リンクを指定するが、この URI は予め動的ページ生成機構によって視聴者の認証情報をその中に含めた URI が記述されている。この URI を視聴者のメディアクライアントよりコンテンツサーバに対して送信する。コンテンツサーバはその URI 情報に基づいてコンテンツを取得するためにファイルへのアクセスを行う。ファイルシステムがこのファイルアクセスを検知すると、URI 内に埋め込まれた認証情報を基にマスター認証サーバに対して視聴者認証を依頼し、マスター認証サーバにおいてデータベースを基に視聴者の認証を行う。その認証の内容とコンテンツに対して設定された視聴条件を比較し、視聴の可否を通知する。また、コンテンツ拡散防止を目的にコンテンツホルダーのみがコンテンツを所持する設定としていた場合には、該当コンテンツの要求部分をシンジケーションに対して送信を開始する。

6.5 コンテンツ取得の手順

視聴者がコンテンツを取得する場合の動作について「Figure 6.4 コンテンツ取得手順シーケンス」に基づいて説明する。

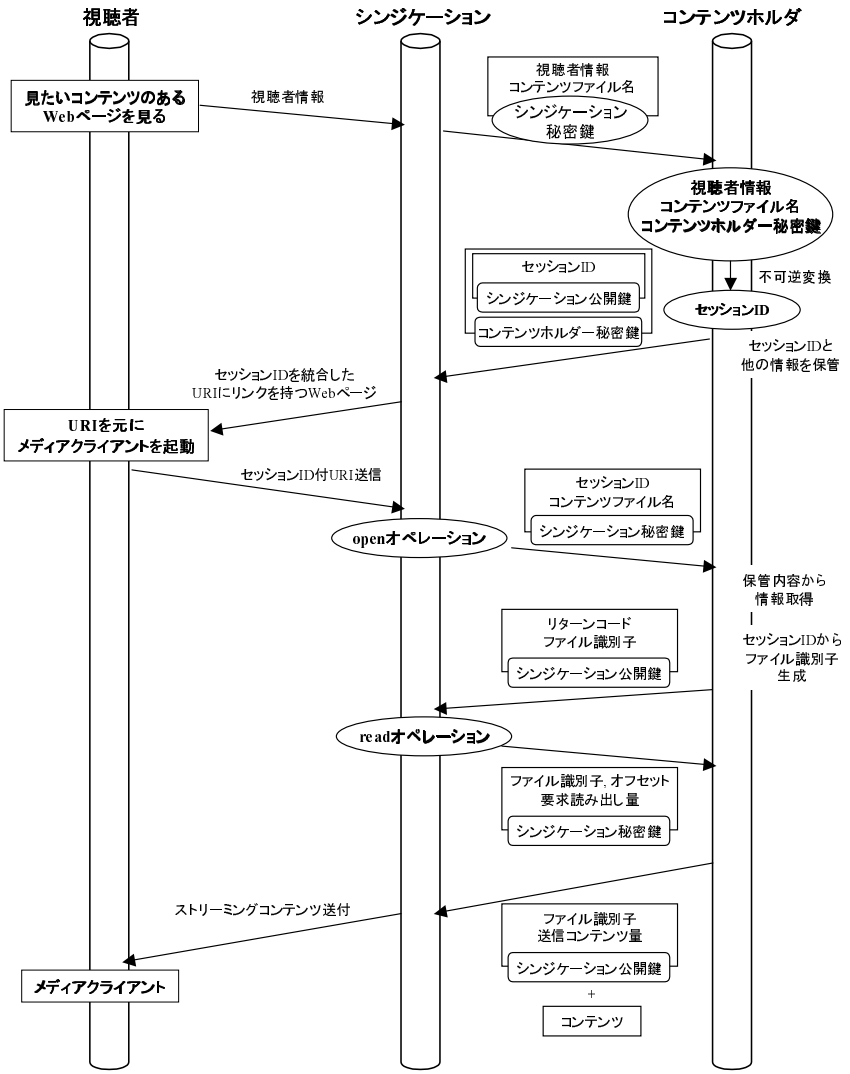

Figure 6.4: コンテンツ取得手順シーケンス

6.5.1 URI 作成プロセス

まず、視聴者があるコンテンツを視聴するために Web ページ を見る。そこで視聴者は Web の認証や SSL(Secured Socket Layer) によるクライアント認証などにより、誰であるかが確認される。この情報より目的のコンテンツを取得するための URI 生成を行うが、これらの情報だけではコンテンツホルダーにとってシンジケーションのなりますましや、再送攻撃に対応できないため、予めこの情報をコンテンツホルダーによってサインすることで対応する。

まずシンジケーションが得た情報より、シンジケーションは視聴者に対して一意な文字列 (視聴者 ID) を作成し、それとファイル名を併せてシンジケーションの秘密鍵を使用し暗号化する。暗号化したものをコンテンツホルダーに送信する。

コンテンツホルダーホストにおいて、シンジケーションの公開鍵を利用して平文化し、視聴者 ID とファイル名を得る。視聴者 ID とファイル名・現在時刻、それにコンテンツホルダーの秘密鍵を併せたものに対して不可逆な変換を行い、これをこのセッションの ID として扱う。

上で生成されたセッション ID をシンジケーションの公開鍵とコンテンツホルダーの秘密鍵で 2 重に暗号化し、シンジケーションに送り返す。

コンテンツホルダーではこのセッション ID と視聴者情報・ファイル名情報を暫定的に保管しておく。

シンジケーションは自らの秘密鍵とコンテンツホルダーの公開鍵と用いてセッション ID を得ることができる。このセッション ID を URI に統合し、視聴者に提供する Web ページ にリンクとして埋め込む。

6.5.2 ストリーミングコンテンツ提供プロセス

視聴者は Web ページ に埋め込まれた URI を指定することで、メディアクライアントを起動し、コンテンツサーバに接続する。

コンテンツサーバはシステムに対し URI が示すファイルの取得を依頼する。URI 中にはセッション ID が入っているため、シンジケーションのシステムはセッション ID を知ることができる。

このセッション ID をコンテンツホルダーの公開鍵で暗号化し、コンテンツホルダーに送信する。ここで、コンテンツホルダーは有効なセッション ID が返却されて来たことで、シンジケーションと視聴者の関係が信用できるも

のであると理解できる。

コンテンツホルダーはこのセッション ID とファイル名、「6.5.1 URI 作成プロセス」において保管された情報より、目的のコンテンツを特定する。ここで、コンテンツホルダーはセッション ID を元に目的のコンテンツのファイル識別子を生成する。

当該コンテンツの提供可否を判断の上、ファイル識別子とリターンコードをシンジケーションの公開鍵で暗号化してシンジケーションに返す。以後、このファイル識別子をコンテンツの特定に用いる。

コンテンツ内容の取得依頼は、シンジケーションからファイル識別子と読みだし開始位置と読み出し長をシンジケーションの秘密鍵で暗号化したものを利用して行う。

6.5.3 セッション ID の有効性

上述のように、セッション ID は、視聴者情報とコンテンツファイル名、またコンテンツホルダー秘密鍵を併せて不可逆変換することで得る。このため、セッション ID を生成するためには、コンテンツホルダーの秘密鍵を知る必要がある。このためコンテンツホルダーにおいてこのセッション ID が有効と判断できれば、そのセッション ID は正しくコンテンツホルダーで処理されたものであると確認できる。

このセッション ID は視聴者情報と目的のコンテンツファイル名を内包するため、もし盗聴されたとしても全てのコンテンツ一般に有効になるわけではない。このため、万一セッション ID が盗聴されたとしても当該コンテンツのみの不正視聴に抑えられる。

またセッション ID とファイル識別子を別にしたことで、平文のまま流される部位のあるセッション ID を盗聴されたとしても悪用される可能性を低く押えられる。

6.6 ソフトウェア

ここでは、本システムを構成する為に必要な各ソフトウェアコンポーネントについて各個に説明する。

全体の構造を、「Figure 6.5 システムの構成」に示す。

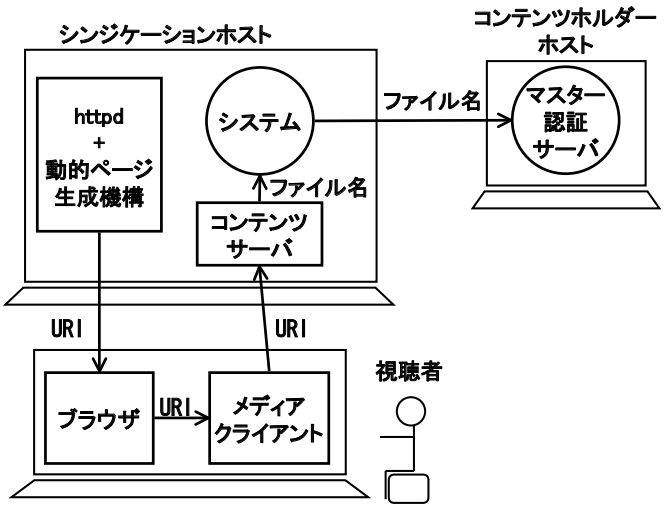

Figure 6.5: システムの構成

6.6.1 コンテンツホルダーホストで動作するマスター認証サーバ

コンテンツホルダーの運営するホストにおいて動作する認証サーバであり、実際の視聴者の認証作業はこのサーバで行う。予め各コンテンツについて設定されたコンテンツ提供ポリシに基づき、視聴ログ等を検索の上コンテンツ提供の可否、もしくは他コンテンツの提供を判断する。

また、コンテンツ提供ポリシに応じ、「6.6.3 シンジケーションで動作するファイルシステム」に対してコンテンツの内容を送信する作業を行う。

6.6.2 シンジケーションで動作するコンテンツサーバ

これは、視聴者に対してコンテンツサービスをする為のコンテンツサーバである。本システムにおいて、このコンテンツサーバは「6.1 モデル設計の前提」に述べた理由により、既存のソフトウェアを使用する。例えば、RealServer や MobileMotion 等の配信で使用される httpd 等がこれに相当する。

6.6.3 シンジケーションで動作するファイルシステム

シンジケーションにおいて、コンテンツサーバからのファイルアクセスに応じて認証機構を動作させる必要があるが、これをファイルシステムとして

設計する。コンテンツサーバからのファイルオープンの要求あれば、その内容に応じ「6.6.1 コンテンツホルダーホストで動作するマスター認証サーバ」で説明したマスター認証サーバに視聴者認証の依頼を行う。

6.6.4 コンテンツホルダーとシンジケーション間の通信

「6.6.3 シンジケーションで動作するファイルシステム」で説明したファイルシステムとマスター認証サーバ間では、視聴者の認証情報・コンテンツの内容・視聴可否についてのエラーコードが通信される。

まずコンテンツサーバからファイルシステムへのファイルアクセス要求に基づき、視聴者の認証情報がマスター認証サーバへ送られる。マスター認証サーバはその視聴者認証情報に応じた処理を行った後、視聴可否情報等についてシンジケーションに通知する。

更に、当該コンテンツの配布ポリシがシンジケーションに対するコンテンツコピーを認めていなかった場合、コンテンツ内容についてファイルシステムからの要求に応じてマスター認証サーバは当該コンテンツの内容を逐次送信する。

6.6.5 シンジケーションにおける動的ページ生成機構

視聴者情報を特定するための情報を、シンジケーションファイルシステムに知らせる必要があるが、それを URI として含めておく。この機構は URI に視聴者情報を埋め込む部分を担当している。

具体的には、「6.5.3 セッション ID の有効性」をで述べたセッション ID をファイル名の中に埋め込む作業を行う。

6.7 問題点の解決

本システムを実現することにより、「2.4 現状の問題点」で挙げた問題点は以下のように解決できる。

6.7.1 視聴ログの収集

視聴者がコンテンツサーバに対してアクセスを行うと、コンテンツサーバはコンテンツホルダーのホスト上で動作しているマスター認証サーバに対して視聴者認証の依頼を行う。この際に、マスター認証サーバに対して提供される情報は、目的のコンテンツ名・視聴者の認証情報であるため、それに視

聴時間等を付加することで、コンテンツホルダーの下に正しく視聴者の視聴ログを残すことができる。また、シンジケーションにおいても同様の情報が流れるため、これを同様の視聴ログとして保管することができる。

6.7.2 メディアに依存しない認証機構

どのようなコンテンツサーバであっても、コンテンツサービスを行う際にはファイルアクセスを行う必要がある。そのファイルへのアクセスという基本的な操作を行う際に、認証作業を行う機構を動作させるため、特定のコンテンツサーバの機能に頼ることなく認証機構を扱うことができる。

これによって、例えば帯域の異なる視聴者に対して異なるメディア技術を用いた運用形態をしているコンテンツ提供形式であっても、「6.7.4 コンテンツ提供ポリシ表現の柔軟性確保」で説明される様なコンテンツ提供ポリシの記述ができる。

6.7.3 コンテンツの拡散抑制

ストリーミングコンテンツの提供は、一般的に一度の視聴においてオリジナルデータの全部をメディアクライアントに対して提供はしない。ほぼ全てのストリーミングメディアで、視聴者とコンテンツサーバ間の帯域に応じてオリジナルコンテンツの一部を送信する形式を採用している。このため、コンテンツサーバと視聴者のメディアクライアント間の通信を全て取得することが可能であったとしてもそのオリジナルデータの復元をすることはできない。

本システムでは、シンジケーションで運用しているコンテンツサーバからのファイルリードアクセス要求に応じて、コンテンツホルダーが持つオリジナルデータから必要部分を提供するモデルを利用する。コンテンツサーバのファイル取得要求は、コンテンツサーバと視聴者のメディアクライアントとの通信内容と相関するため、その通信内容がオリジナルコンテンツの全てでないのと同様に、ファイル取得要求もオリジナルコンテンツ全てを要求することはない。これにより、シンジケーションを通過するデータはオリジナルデータの全てではないため、オリジナルデータを再構成することはできないこととなる。

6.7.4　コンテンツ提供ポリシ表現の柔軟性確保

視聴ログがマスター認証サーバにデータとして残っているため、それを利用してコンテンツ視聴条件を動的に変化させることができる。例えば、あるコンテンツを視聴後でないと該当コンテンツを視聴することができないようにすることや、一定時間以内に提供するコンテンツ数の制限などを行うことができるようになる。

教育環境においてはしばしば「ある講義を履修済であることを条件にこの講義を受講することができる」というような条件付けを行うことがある。現状の認証構造では、これを Web ページにおける認証と CGI 等の動的ページ生成機構を用いて、コンテンツに対する URI リンクを提供しないことで実現するが、本システムにおいては提供されるコンテンツ自身の認証を加えることで、更に強固な制限ができる。

Chapter 7

実証ソフトウェアの設計と実装

ここでは、本モデルを実証する上でのソフトウェアの実装について議論する。まずシステムの部品構成を説明し、更にその連係の方法について説明する。

7.1 システムの構成

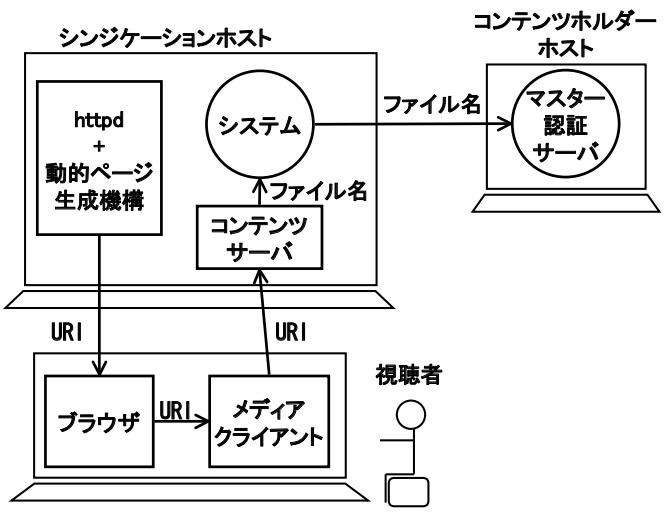

Figure 7.1: システムの構成

本システムは、大きく分けて、マスター認証サーバ・シンジケーションにおけるファイルシステム・シンジケーションにおいて URI を提供する動的ページ生成機構の 3 つからなる。

マスター認証サーバはコンテンツホルダーが管理するサイトで動作し、視聴者の認証・コンテンツ提供ポリシの実現・コンテンツ提供を担当する。

シンジケーションにおけるファイルシステムは、コンテンツサーバからのファイル要求を取得し、マスター認証サーバに視聴者認証とコンテンツ提供の可否伺いを転送する。そしてその結果をコンテンツサーバに提供する。また、マスター認証サーバから送られるコンテンツ内容をコンテンツサーバに提供する。

シンジケーションにおいて URI を提供する動的ページ生成機構は、URI のファイル名部分に視聴者情報を付与する。

7.2 実装環境

本システムは「Table 7.1 実装環境」に示す構成の環境で動作するよう実装を行った。

OS	Linux 2.2.17 + glibc-2.1.3
DataBase	PostgreSQL 7.0.2
httpd	Apache 1.3.14 + php-3.0.18-i18n-ja
公開鍵暗号系	OpenSSL-0.9.6

Table 7.1: 実装環境

また、Solaris や FreeBSD などを用いた環境においても動作することが確認されている。

7.3 マスター認証サーバの実装

ここで、コンテンツホルダーにて運用される、マスター認証サーバの実装について説明する。マスター認証サーバはサーバデーモンとして実装され、視聴者の認証とコンテンツ提供の可否判断、コンテンツの送信を担当する。

7.3.1　ネットワーク通信部

マスター認証サーバはシンジケーションからのリクエストが無い限りデーモンとして待機状態にある。シンジケーションから認証要求の為の通信が行なわれると、マスター認証サーバは fork によって分岐し、子プロセス側が「7.4.2 セッションマネージャ」で説明するシンジケーションのセッションマネージャとの通信を開始する。親プロセス側は引続きデーモンとして待機状態になる。

子プロセス側は視聴者の認証からコンテンツ提供の終了までセッションマネージャとの通信を確保し、セッションマネージャからの要求に応える。

7.3.2　データベースインタフェース

マスター認証サーバはコンテンツ情報やシンジケーション情報などを PostgreSQL を用いた RDBMS に保持している。このデータベースサーバと通信する為に libpq という API(Application Program Interface) を用いた。

データベースに入れられる情報の属性定義などについては付録 B「12.1 データベーステーブル：コンテンツホルダー」に示す。

7.3.3　暗号系

ユーザ認証に付いては、公開鍵暗号系の暗号化・平文化が必要になるが、これを OpenSSL に付属の RSA 実装を利用した。

7.4　シンジケーションファイルシステム

本システムにおいて、シンジケーションでファイルシステムとして動作するプロセスは、ファイルシステムとのインタフェースとして動作するデーモンプロセスと、マスター認証サーバとの通信を管理するセッションマネージャプロセスの二つである。これら二つのプロセスが協調して動作し、コンテンツサーバに対してファイルサービスを提供する。

デーモンプロセスとセッションマネージャプロセス、コンテンツホルダーとの関係を「Figure 7.2 シンジケーションファイルシステムの構成」に示す。

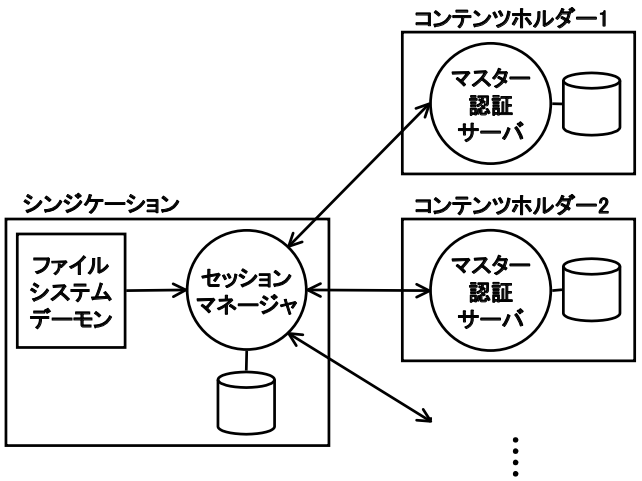

Figure 7.2: シンジケーションファイルシステムの構成

7.4.1 ファイルシステムデーモン

ファイルシステムデーモンは Sun NFS のプロトコルを利用した RPC 待ち受け部分である。ファイルシステムデーモンは、Sun NFS プロトコルを通じてコンテンツサーバからのファイルアクセスを受け付け、その要求に応じてセッションマネージャと通信を行う。この部分はファイルのアクセスコントロール等の操作は一切行わない。

7.4.2 セッションマネージャ

セッションマネージャはファイルシステムデーモンを通じてコンテンツサーバのファイルアクセスを受け付け、その要求内容に応じてマスター認証サーバに要求を転送する。

セッションマネージャは、「Figure 7.2 シンジケーションファイルシステムの構成」に示されるように、ファイルアクセス要求に応じて同時に複数のマスター認証サーバとの通信を確保できるように実装した。

シンジケーションファイルシステムにおいて二つのプロセスを使用して実装した理由は、ファイルシステムデーモンを Sun NFS(Sun Microsystems Network File System) プロトコルを利用して実装を行っていることに起因する。Sun NFS プロトコルを利用するためには RPC(Remote Procedure Call)

を用いて実装を行わなくてはならないが、処理が RPC の待ち受けルーチンにある場合に、通信の待ち受けを同時に行なうのは難しい。これを解決する為、RPC 待ち受けをするプロセスと、その他の通信の待ち受けするプロセスに分けて実装した。

7.4.3 データベースインタフェース

セッションマネージャもマスター認証サーバと同様に、コンテンツホルダー情報やコンテンツのキャッシュなどを PostgreSQL を用いた RDBMS に保持する。同様にデータベースサーバと通信する為に libpq を用いて実装を行った。

データベースに入れられる情報の属性定義などについては付録 B 「12.2 データベーステーブル：シンジケーション」に示す。

7.5 動的ページ生成機構

動的ページ生成機構では、Web ブラウザから送られる SSL クライアント認証を用いた証明書、もしくは httpd によるユーザ認証を元にした視聴者情報を元に、本システムで用いる一意な視聴者情報を生成する。

この一意な視聴者情報をコンテンツホルダーに送信すると、コンテンツホルダーは視聴者情報とファイル名・コンテンツホルダーの秘密鍵を合わせた情報を MD5 を用いて不可逆変換し、それをセッション ID として動的ページ生成機構に返却する。

動的ページ生成機構は受け取ったセッション ID を、「7.7 ファイル名」で示される形のように URI で示されるファイル名の中に埋め込み、ブラウザに対してそのリンクを提供する。

本システムでは、動的ページ生成機構として PHP を用いた。

ただし、公開鍵暗号系を用いた暗号化・平文化については PHP で対応していなかったために外部プログラムとして実装した。

7.6 通信プロトコル

シンジケーションとコンテンツホルダー間の通信に関するプロトコルについては、各フォーマットを付録 A「11 通信プロトコル」に示す。

なお、各データはネットワークバイトオーダで通信される。

7.7　ファイル名

本システムでアクセスされるファイル名は、視聴者識別情報より生成されたセッション ID を含む。

	[マウントポイント]/[コンテンツホルダー識別名]/[セッション ID]/[コンテンツホルダーが識別するファイル名]
例	/mntfs/KEIO_CS/2adca4a80228060daf328961e27d0473/contents/start.rm

Table 7.2: URI 中のファイル名構成

具体的には、「7.2 URI 中のファイル名構成」に示すように、本システムのファイルシステム部をマウントしたディレクトリ名の直下にコンテンツホルダーを識別するための情報をディレクトリ名であるかのようにを記述する。その次に、セッション ID を同様に記述する。最後にコンテンツホルダーがコンテンツを特定するためのファイル名を記述し、一つのファイル名として構成する。

7.8　ファイルアクセスに対する応答

コンテンツサーバを通さずに、管理目的やコンテンツ名検索目的で直接ファイルを操作する場合がある。本システムでは、ファイルシステムとして実装されているため、このような場合にも応答する必要がある。

直接ファイル操作を行う場合にどのようなファイルツリーになるかを「Figure 7.3 ファイル操作時のファイルツリー」に示した。

「7.7 ファイル名」で示した、セッション ID が入るべき部分に、セッション ID として 0 という数字を入れたファイルへのアクセスを行う。このことで、コンテンツホルダーは正規の手順を踏んでアクセスされた物ではないことが判別できるため、相応の対応が可能となる。

なお、直接ファイル操作をする際には、opendir, readdir, fstat の操作のみがサポートされる。ここではファイルの読み込み操作を行うことはできない。

この機能を実装することで、シンジケーションはコンテンツ名の一覧を取得することができるようになる。

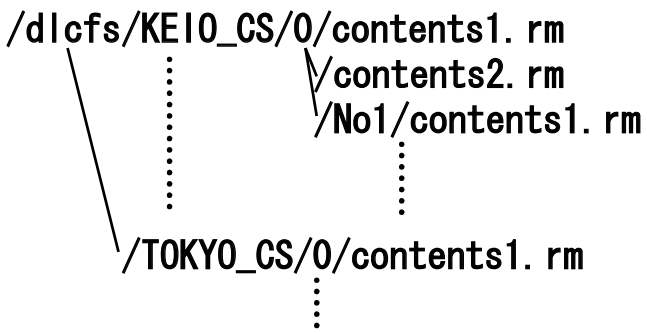

Figure 7.3: ファイル操作時のファイルツリー

Chapter 8

評価

ここで、「2.4 現状の問題点」で挙げた問題に対し本システムが解決すると予想された部分について、評価する。まず、結果を他研究と比較したものを提示した後に、本モデルにおける問題点充足について述べる。

8.1 他研究との比較

「Table 8.1 問題点充足に関する比較」に、本システムが他の関連研究と比較してどの程度問題点を充足しているのかを比較したものを挙げる。

	ポリシ柔軟性	ログ集中	コンテンツ拡散防止	メディア非依存
本システム	○ [1]	○ [2]	○	○
mirror	×	×	×	○
分散ファイルシステム	×	△ [3]	△	○
RealServer + RealProxy	◎	◎	○	×
WMRM	◎	△	△	△ [4]

Table 8.1: 問題点充足に関する比較

8.2 コンテンツ提供ポリシ表現の柔軟性欠如への対応

本システムにおいて、コンテンツ提供ポリシはデータベースの関係記述をすることで提供され、結果はファイルアクセスの可否、もしくは提供コンテンツの変化で表される。

リレーショナルデータベースにおける関係記述は、他のデータを参照して結果を返すことができるため、例えば過去の視聴ログを元にアクセスコントロールを行うことができるようになった。

これによって、ユーザ名・パスワードの対でアクセスの可否を判断する形式などに比べてコンテンツ提供ポリシ表現の柔軟性は確保できたといえる。

8.3 分散してしまう視聴ログへの対応

本システムでは、視聴ログについて以下の項目についてはコンテンツホルダー・シンジケーション共に取得できる。このため、コンテンツホルダーはシンジケーションに対するアクセスであってもその視聴ログを取得することができる。

- コンテンツ名
- コンテンツ取得開始日時
- コンテンツ取得終了日時
- 視聴者
- 取得可否に関するエラーコード

しかし、以下の項目についてはシンジケーションのみで記録される。

- メディアクライアントの識別 ID(RealMedia 等の場合)
- 再送の有無
- 通信の障害等に関するエラーコード

[1]「8.3 分散してしまう視聴ログへの対応」で述べた、情報が不足のため通信状況に対する制限に対応し切れない可能性がある

[2]得られるはずの全ての情報を得ていないため

[3]ログファイルも分散ファイル上で共通のものを使用することで各サーバのログを統合することはできる

[4]他のメディアでも対応できるよう API を用意しているが、Windows 環境のみでの対応である。

また、以下の項目については結果を推測できる可能性はあるが、明示的には不明である。

- 送信量
- コンテンツ取得速度
- 取得されたメディアのプロトコル

上記いずれのものも、コンテンツサーバからのログには出力される可能性のあるものである。

これにより、本研究は、多くの場合のコンテンツ提供ポリシ設定には対応できるものの、特に通信状況に関する制限等には対応しきれない可能性がある。

これは、帯域の幅に応じて別のストリーミング技術を利用したコンテンツ視聴の勧告などの制御はできないことを意味する。

8.4 コンテンツの拡散への対応

本システムにおいては、「2.1.1 ストリーミング技術の特徴」で述べた特徴である、コンテンツの通信に際して全ての内容を送信するわけではないという特性を根拠にコンテンツ拡散への安全性を確保している。すなわち、シンジケーションにおいて完全な形のコンテンツを取得することはできない。

しかし、同じコンテンツを多くの帯域で多くの回数視聴された場合において、完全なコンテンツとして取得されない可能性を否定できるものではない。

8.5 認証運用のメディア依存性への対応

本システムは、コンテンツを提供する際に行われるファイルシステムへの実装を行っている。このため、コンテンツ提供の際にファイルシステムへのアクセスが行われるコンテンツサーバに関しては、正常に動作することが確認できた。

しかしコンテンツ提供の際に、コンテンツ自体をデータベースから取得するものもある。Oracle interMedia[17] のようなものがそれにあたる。検証の結果、このような技術には対応できないことが確認された。また、Windows 上から提供するコンテンツサーバには基本的には対応できない。

8.6　問題点充足のまとめ

ここで、「2.4 現状の問題点」で提示した 4 つの問題点について、本モデルがどの程度問題充足したかを評価した。

以上の評価より、本モデルは各論において他の研究に劣るものがあるものの、総体的には本モデルが優位にあると言える。

Chapter 9

結論

9.1 本研究のまとめ

本システムは、コンテンツホルダーとシンジケーションが独立した組織で運営されている場合に現出する、下記に挙げる 4 点の問題点を総合的に解決するためのモデルを考察した。

- コンテンツ提供ポリシ表現の柔軟性欠如
- 分散してしまう視聴ログ
- コンテンツ拡散の危険性
- 認証運用のストリーミングメディア依存性

また、このモデルが上記問題点をどの程度充足するか検証をするためのソフトウェアを設計、実装した。このソフトウェアを評価した結果、それぞれ部分においては他の技術に劣る部分が存在するものの、総体的に優位にあることを示せた。

特に、条件変更の動的変化に対応するコンテンツ提供ポリシの設定をストリーミングメディアに依存せずに行えることは、他の研究と比較して優位にあると言える。

9.2 今後の研究課題

今回の設計においては、コンテンツ提供ポリシを反映できるようにすることをそのものを重視した設計になっているため、提供パフォーマンスが犠牲になっている。このため、今後は提供パフォーマンスの向上を考慮した設

計にすることが考えられる。これは UDP/IP を用いた通信を行うこと、マルチキャストに対応させること、コンテンツ拡散に対して安全なコンテンツキャッシュの開発を行うこと等が含まれる。

また、アプリケーションを開発する際に、ファイルシステムを通さず本システムを利用できるように、API を提供することも考えられる。これによって、現状の本システムでは取得できなかった、「8.3 分散してしまう視聴ログへの対応」で述べた帯域の問題に関する情報や、メディアクライアント固有 ID の取得などの情報を取得・応用することができるようになる。

更に視聴者が認証用の専用クライアントを用いることで、視聴者認証の精度を向上させることも考えられる。

今後、このような方向性に基づいて引続き開発を行う。

Chapter 10

謝辞

本論文をまとめるにあたり、多くの方の協力を得ました。特に多大なご指導くださいました、慶應義塾大学政策メディア研究科 村井 純 教授、同環境情報学部 楠本 博之 助教授、北陸先端科学技術大学院大学情報科学研究科 篠田 陽一 助教授の各氏に深く感謝致します。

また、本論文の方向性について示唆をくださり、その後も多く助言を頂きました、慶應義塾大学 SFC 研究所の 植原 啓介氏、大川 恵子氏に感謝致します。

本論文の構成方法やその内容について多くの助言をくださった、ヴィジョンアーツ (株) の 五十嵐 洋介氏、(株) 東芝 の 東 秀明氏、業務時間の一部を論文執筆に割くことを許して頂き快くオフィスの使用を許可頂きました、ヴィジョンアーツ (株) に深く感謝の意を表します。

また、助言や協力を頂きました、政策メディア研究科 の 小浦 大将氏、三谷 洋司君、学生コンサルタント後輩 の 尾崎 祥子さん、吉本 智子さん、徳田村井楠本中村共同研究室の皆さん、その他執筆に配慮してくださった皆様には公私共々お礼を申し上げると共に、感謝致します。

Chapter 11

付録 A: 通信プロトコル

ここで、コンテンツホルダーとシンジケーション間の通信プロトコルについて示す。

11.1 プロトコル共通ヘッダ

通信段階識別子	2bytes
将来の拡張用	6bytes
通信長	8bytges
通信内容	n bytes ($n < 16^8$)

Table 11.1: プロトコル共通ヘッダ

コンテンツホルダーとシンジケーション間の通信プロトコルは全て「Figure 11.1 プロトコル共通ヘッダ」に示される形のヘッダ構造を持つ。

この中で、通信段階識別子に入る値を「Figure 11.2 通信段階識別子」に示す。

11.2 CREATE_URI_R 時の通信内容

シンジケーション中で一意な視聴者情報と、要求されるコンテンツファイル名をコンテンツホルダーに送る。

具体的なフォーマットを「Figure 11.3 CREATE_URI_R 時のフォーマット」に示す。

0	PING	相手の動作確認 (データは無し)
1	PONG	動作確認応答 (データは無し)
4	CREATE_URI_R	URI 作成時にシンジケーションからコンテンツホルダーに送られる要求
5	CREATE_URI_A	上記応答
8	OPEN_SESSION_R	ファイルの open オペレーションが行われる際にシンジケーションからコンテンツホルダーに送られる認証要求
9	OPEN_SESSION_A	上記応答
12	CONTENTS_READ_R	コンテンツを取得する際にシンジケーションからコンテンツホルダーに送られるコンテンツ送信要求
13	CONTENTS_READ_A	上記応答

Table 11.2: 通信段階識別子

視聴者情報データ長	4bytes
将来の拡張用	4bytes
コンテンツファイル名長	8bytes
視聴者情報	n bytes ($n < 16^4$)
コンテンツファイル名	n bytes ($n < 16^8$)

Table 11.3: CREATE_URI_R 時のフォーマット

このデータは、シンジケーションの秘密鍵で暗号化した上、共通ヘッダとともに送信される。

11.3 CREATE_URI_A 時の通信内容

視聴者情報と、コンテンツファイル名、コンテンツホルダー秘密鍵を MD5 を用いて不可逆変換したセッション ID をシンジケーションに送る。セッション ID は 16bytes となり、一意であると推測できる。

このデータはシンジケーション公開鍵とコンテンツホルダー秘密鍵を用いて暗号化され、共通ヘッダとともに送信される。

11.4 OPEN_SESSION_R 時の通信内容

コンテンツサーバがファイルオープンをする際に、セッション ID とコンテンツファイル名をシンジケーションの秘密鍵を用いて暗号化したものを、コンテンツホルダーに送る。

このときの具体的なフォーマットを「Figure 11.4 OPEN_SESSION_R 時のフォーマット」に示す。

セッション ID	16bytes
コンテンツファイル名長	8bytes
コンテンツファイル名	n bytes ($n < 16^8$)

Table 11.4: OPEN_SESSION_R 時のフォーマット

11.5 OPEN_SESSION_A 時の通信内容

コンテンツサーバのファイルオープンの返り値として、リターンコードとファイルの識別子をシンジケーション公開鍵で暗号化したものをシンジケーションに返却する。

このときの具体的なフォーマットを「Figure 11.5 OPEN_SESSION_A 時のフォーマット」に示す。

リターンコード	4bytes
将来の拡張用	4bytes
ファイル識別子	32bytes

Table 11.5: OPEN_SESSION_A 時のフォーマット

11.6 CONTENTS_READ_R 時の通信内容

これは、シンジケーションがコンテンツホルダーに対してコンテンツの内容を要求する際に送られる。このときの具体的なフォーマットを「Figure 11.6 CONTENTS_READ_R 時のフォーマット」に示す。

ファイル識別子	32bytes
読み出しオフセット	8bytes
要求読み出し量	8bytes

Table 11.6: CONTENTS_READ_R 時のフォーマット

これらをシンジケーション秘密鍵で暗号化して送信される。

11.7 CONTENTS_READ_A 時の通信内容

コンテンツホルダーがシンジケーションに対してコンテンツ内容を送る際の具体的なフォーマットを「Figure 11.7 CONTENTS_READ_A 時のフォーマット」に示す。

ファイル識別子	32bytes
送信コンテンツ量	8bytes
コンテンツ内容	n ($n < 16^8$)

Table 11.7: CONTENTS_READ_A 時のフォーマット

これらをシンジケーション公開鍵を利用して暗号化し、送信する。

ただし、ここでは計算機資源の節約のために、暗号化する部分はファイル識別子と送信コンテンツ量に限られる。コンテンツ内容そのものについては、

平文のまま送信する。

11.8 エラーコード

ファイルオープンをした際に、返り値として視聴可否の判定やファイル操作の結果を返す。そのときのエラーコードの一覧を「11.8 プロトコル : エラーコード」に示す。

正常系	OKEY	視聴可
	OKEY_CACHE	視聴可, シンジケーションにおけるキャッシュを許可する.
異常系	NOT_A_DIRECTORY	ディレクトリオペレーションをされた相手がディレクトリでない.
	IS_A_DIRECTORY	ファイルオペレーションをされた相手がディレクトリである.
	BAD_REQUEST	ファイル操作要求が不正である.
	STALE_ID	ファイル ID の有効期限切れ, もしくは不正
	FORBIDDEN	該当コンテンツが無い
	SERVER_ERROR	サーバエラー
拒否系	PERMISSON_DENY	視聴拒否 (一般)
	TOO_MANY_USERS	
	DO_NOT_HAVE_	

Table 11.8: プロトコル : エラーコード

Chapter 12

付録 B: データベーステーブル

ここでは本システムで使用されるデータベーステーブルの概要を挙げる。

12.1 データベーステーブル：コンテンツホルダー

属性名	型	説明
sin_id	varchar(8) (key)	シンジケーション管理用 ID
sin_name	text	管理者が識別するために用いるシンジケーション名[1]
hostname	text	インターネット上のシンジケーションホスト名[2]
sin_cert	text	シンジケーションの公開鍵

Table 12.1: コンテンツホルダー：シンジケーションリストテーブル

12.1.1 シンジケーションリストテーブル

シンジケーションリストテーブルは、コンテンツホルダーがコンテンツを提供する可能性のあるシンジケーションをリストする。シンジケーションによって署名された視聴者情報が正しく視聴者の物であるかを確認するための、シンジケーションの公開鍵もここに保管する。

[1]本システムは関与しない
[2]これは FQDN(Fully Qualified Domain Name)、IPv4 アドレス、IPv6 アドレスのいずれでも良い

属性名	型	説明
fid	int (key)	アクセスされたコンテンツのファイル ID
dirid	int (key)	アクセスされたコンテンツのディレクトリ ID
con_name	text	アクセスされたコンテンツのファイル名
sin_id	varchar(8)	コンテンツ要求元のシンジケーション ID
student_id	varchar(16)	コンテンツ視聴者の視聴者番号
readtimestamp	datetime	アクセス開始日時
return_code	int	提供時のエラーコード

Table 12.2: コンテンツホルダー：視聴ログクラス

属性名	型	説明
fid	int (key)	該当タプルを意味するファイル ID
dirid	int (key)	該当タプルが所属するディレクトリ ID
pdirid	int	親ディレクトリのディレクトリ ID
stat	int	ファイルステータス
con_name	text	コンテンツ名
filename	text	該当コンテンツを意味するファイルのフルパス

Table 12.3: コンテンツホルダー：コンテンツの情報クラス

12.1.2 コンテンツの情報クラス

コンテンツホルダーが管理するコンテンツ情報は、仮想的にファイルツリーをデータベース上に構築する。dirid が 0 かつ pdirid が 0 のタプルを、提供するファイルツリーのルートとして定義する。各ファイルは、fid と dirid をキーとして識別される。

属性名	型	説明
fid	int (key)	該当タプルを意味するファイル ID
dirid	int (key)	該当タプルが所属するディレクトリ ID
sin_id	varchar(8)	相手シンジケーション ID
watcher_id	varchar(16)	視聴者 ID
policy_id	int	該当コンテンツに適用する提供ポリシ

Table 12.4: コンテンツホルダー：提供ポリシ選択クラス

12.1.3 コンテンツ提供ポリシ選択クラス

コンテンツ提供ポリシは、ファイル毎もしくはディレクトリ毎に記載することができる。fid 0 はディレクトリを意味する。

watcher_id が NULL の場合は、相手シンジケーション全てに該当する場合に適用する提供ポリシを意味する。sin_id が同じで、 watcher_id が NULL のタプルと該当視聴者の watcher_id が記載されているタプルが存在する場合には、後者を採用する。

相手シンジケーション ID が NULL の場合は、シンジケーションに関わらず適用する提供ポリシを意味する。

属性名	型	説明
policy_id	int (key)	コンテンツ提供ポリシ記述 ID
policy	text	コンテンツ提供ポリシ [3]

Table 12.5: コンテンツホルダー：コンテンツ提供ポリシ記述クラス

[3]SQL 文で記述され、これが真であればコンテンツ提供が許可される

属性名	型	説明
policy_id	int (key)	コンテンツ提供ポリシ記述 ID
policy	text	コンテンツ提供ポリシ

Table 12.6: コンテンツホルダー：契約記述クラス

12.2 データベーステーブル：シンジケーション

属性名	型	説明
con_id	varchar(8) (key)	コンテンツホルダー管理用 ID
con_name	text	管理者が識別するために用いるコンテンツホルダー名[4]
hostname	text	インターネット上のマスター認証サーバホスト名[5]
con_cert	text	コンテンツホルダーの公開鍵

Table 12.7: シンジケーション：コンテンツホルダーリストクラス

12.2.1 コンテンツホルダーリスト

これは、シンジケーションがコンテンツホルダーのホストを識別する為のデータベースである。また、同時にコンテンツホルダー認証用に公開鍵を保管しておく為の属性を用意してある。

属性名	型	説明
watcher_id	varchar(16) (key)	視聴者番号 (学籍番号等に相当する)
student_cert	text	一意な視聴者情報

Table 12.8: シンジケーション：視聴者リストクラス

[4]本システムは関与しない

[5]これは FQDN(Fully Qualified Domain Name)、IPv4 アドレス、IPv6 アドレスのいずれでも良い

属性名	型	説明
con_id	varchar(8)	要求コンテンツのコンテンツホルダー ID
fid	int (key)	アクセスされたコンテンツのファイル ID
dirid	int (key)	アクセスされたコンテンツのディレクトリ ID
con_name	text	アクセスされたコンテンツのファイル名
watcher_id	varchar(16)	コンテンツ視聴者の視聴者番号
readtimestamp	datetime	アクセス開始日時
return_code	int	提供時のエラーコード

Table 12.9: シンジケーション：視聴ログクラス

12.2.2 視聴ログ

これは、コンテンツの視聴ログを保管する為のデータベースである。基本的にコンテンツサーバから「7.4.1 ファイルシステムデーモン」へコンテンツの要求が行なわれた時点で新しくタプルを挿入する。

属性名	型	説明
con_id	varchar(8) (key)	コンテンツホルダー管理用 ID
fid	int (key)	該当タプルを意味するファイル ID
dirid	int (key)	該当タプルが所属するディレクトリ ID
pdirid	int	親ディレクトリのディレクトリ ID
sequence	int	コンテンツの中の相対位置
contents_body	large object (単位バイト)	コンテンツの中身を単位バイトで区切った部分長だけ保管

Table 12.10: シンジケーション：コンテンツキャッシュクラス

12.2.3 コンテンツキャッシュ

シンジケーションにおいてキャッシュが許可されたコンテンツが提供された場合に、「12.10 シンジケーション：コンテンツキャッシュクラス」へコンテンツの内容を保管する。以後、同じコンテンツに対する要求があった場合にはこのテーブルよりコンテンツを提供する。

Bibliography

[1] Daniel C.Lynch/Marshall T.Rose 編著、村井 純監訳『インターネットシステムハンドブック』
(インプレス出版, 1996)

[2] チャーリー・カウフマン+ラディア・パールマン+マイク・スペシナー著、石橋 啓一郎+菊池 浩明+松井 彩+土井 裕介訳『ネットワークセキュリティ』
(プレンティスホール出版, 1997)

[3] 大澤 光編著『インターネットストリーミング』
(共立出版, 2000)

[4] Edward R Zayas
『AFS Programmer's Reference: Architectural Overview』
(Transarc Corporation, 1991)

[5] Transarc Corporation
『AFS Programmer's Reference: Authentication Server Interface』
(Transarc Corporation, 1993)

[6] 『AFS Frequentry Asked Questions』
http://www.angelfire.com/hi/plutonic/afs-faq.html
(1998)

[7] RealNetworks Inc. 『RealServer7.0 認証機能』
http://www.jp.realnetworks.com/products/servers/userauth.html
(RealNetworks Inc.)

[8] RealNetworks Inc. 『RealProxy(TM) ホワイトペーパー』
http://service.jp.real.com/help/library/whitepapers/rproxy/proxy.html
(RealNetworks Inc., 2000)

[9] J.Kohl 『The Kerberos Network Authentication Service(V5)』
(RFC1510, 1993)

[10] SP Miller, BC Neuman, JI Schiller, JH Saltzer, 『Kerberos Authentication and Authorization System』
(Project Athena technical Plan, Section E.2.1, MIT, December 1987)

[11] Barry Jaspan 『Kerberos Users' Frequently Asked Questions』
ftp://rtfm.mit.edu/pub/usenet/news.answers/kerberos-faq/user
(news:comp.protocols.kerberos, 1995)

[12] 稲村 雄「Kerberos」『最新の暗号技術によるセキュリティの実現』
OpenDesign No.14 (CQ 出版, 1996)

[13] AuthPG Project 『PostgreSQL Authentication Module for Apache』
http://authpg.sourceforge.net/
(2000)

[14] Sony Corporation 『PercasTV』
http://www.percastv.net/
(Sony Corporation, 2000)

[15] Microsoft 『Windows Media Rights Manager』
http://www.microsoft.com/windows/windowsmedia/en/wm7/drm.asp
(Microsoft, 2000)

[16] 郵政省通信政策局『ストリーミング技術等の新技術を用いたコンテンツ流通ビジネスに関する研究会』
http://www.mpt.go.jp/pressrelease/japanese/tsusin/000630j503.html
(郵政省通信政策局, 2000)

[17] Oracle Corporation 『interMedia main page』
http://technet.oracle.com/products/intermedia/listing.htm
(Oracle Corporation, 2000)