

# rtmpdumpによるライブ動画ストリームの予約保存

2009SE265 鈴木 一輝

指導教員：後藤 邦夫

## 1 はじめに

近年、注目を集めているインターネットサービスに、リアルタイムで動画を中継する動画共有サービスがある。動画共有サービスでは、単なる娯楽としての利用の他、政府の記者会見や事業仕分けの中継なども配信され、新聞やテレビを越えるリアルタイム性を有する情報発信媒体として機能している [6].

本研究では、動画中継サービスを行う代表的なサイトである Ustream, Stickam JAPAN!, ニコニコ生放送を例にとり、3つのサイトでの動画中継のプロセスを調べ、中継される予定の動画を予約録画するシステムの構築を目指した。またこのシステムは、3つのサイト以外の動画配信サービスを行っているサイトでも汎用的に利用できる、拡張性の高いシステムへの応用を想定し構築した [7][10][9].

## 2 システムの概要

このシステムでは、配信告知が行われているページの URL と、その予定時刻という2つの情報から、ストリーミング動画の録画予約を行えるようにする。3つのサイトとも、ホームページの構造、通信時の細かい挙動等は異なるが、システム利用者の利便性を考慮し、1つのソフトウェアで3つのサイトからのストリーミング動画のダウンロードに対応出来るようにする。また今後3つのサイト以外のサービスも録画などの機能追加が容易になるように、GUIなしのコマンドで動くシステムとする。

### 2.1 RTMP の概要

本研究で対象となる3つのサイトは、すべて RTMP というストリーミングプロトコルを利用して動画のストリーミング配信を行っている [1]。この RTMP は Adobe 社が開発しているプロトコルであり、Adobe Flash Player, サーバ間の通信環境で動く。

この RTMP によって送られてくるデータは通常 PC 内に保存される事がないため、キャッシュの形で動画の取得は望めない。また Web ブラウザには RTMP サーバの URL は表示されず、履歴にも残らない。よって何らかの形で URL を調べる必要がある。

### 2.2 RTMPDUMP の概要

RTMPDUMP とは、RTMP 方式で配信されているストリーミング動画をダウンロードする事が出来るオープンソースのソフトウェアである [3].

本来 RTMP はストリーミングサーバと Flash Player 間で動作する通信プロトコルなので、ユーザ側からは Flash Player に送られてくる動画データを取得をする必要あるが、RTMPDUMP が擬似的に Flash Player の役割を果たすことによって、直接データを引き出すことが出来る。

RTMPDUMP には、ストリーミング動画の URL や RTMP サーバ名、出力ファイル名など、30以上のオプションが用意されており、サービスを行っているサイト毎に複数のオプションを適時指定して、引数として渡す必要がある。

### 2.3 録画予約に必要な情報の調査法

RTMP サーバの URL をはじめとする動画の録画予約に必要な情報の調査法であるが、既存のストリーミング動画の録画を行うプログラムのソースなどを参考に、proxy サーバとパケットキャプチャの2つを利用し、proxy サーバに squid を、パケットキャプチャに Wireshark をそれぞれ使用する [4][5].

## 3 各サイトの配信システム

ここでは proxy サーバなどを利用して調べたそれぞれのサイトの特徴と、そこから見えてくる動画取得に必要な要素を挙げる。

### 3.1 Ustream の動画取得に必要な情報

まず、Ustream のサーバとユーザ間でどのような通信が行われているのかを詳しく把握するために、Proxy サーバを用意し、proxy サーバを経由させ通信を行うことによってログファイルを作成し、通信記録を確認した結果、amf ファイルのやりとりが行われている事と、amf ファイルの URL を確認した [2]。その中に RTMP サーバの URL が含まれており、次のようになっている。

- `rtmp://[サーバ名]/ustreamVideo/[ustID]`

これで RTMP サーバの URL が判明したので、これを引数とし、さらに `-a`, `-y`, `-f` を追加して RTMPDUMP に渡せば動画の取得が可能だと考え、実行した結果、RTMPDUMP の起動を確認した。

### 3.2 Stickam JAPAN! の動画取得に必要な情報

Ustream と同様に、Proxy サーバを経由させることによって、必要な情報を探そうと試みたが、必要な情報が発見出来なかったため、パケットキャプチャを利用しての通信内容の把握を試みる。全パケットの中で `'rtmp://'` という文字列が含まれている箇所を文字列検索し、さらにそれを加工して RTMP サーバの URL を得た。また amf データに追加文字列を入力する `-C` に適切な引数を渡せば RTMPDUMP が動くことから入力コマンドを決定した。

### 3.3 ニコニコ生放送の動画取得に必要な情報

配信システムを探るために自分のアカウントで実際に配信してみたところ、配信者向けに xml ファイルが取得出来ることが分かった [8]。またパケットキャプチャにより先ほどとは違う xml ファイルを確認した。この2つの xml ファイルの内容を比較するといくつかの相違点が判明した。

- URL 内の publicorigin と liveedge の違い
- getplayerstatus.xml の URL が短い
- 動画番号以降のパラメータの長さ

これらの情報を元に引数を決定したが、その引数で RTMPDUMP を起動させたところ、サイト運営側が流している公式配信に関しては動画の録画に成功したものの、一般の有料会員の放送は録画することが出来なかった。

### 3.4 各配信サイトの共通点と比較

3 サイトの比較から、ログインの必要性の有無によって、取得すべき情報も変わってくる事が変わる事が分かる。つまりログインを必要としていないサイトではサーバ名とユーザ ID の 2 つの情報、ログインを必要とするサイトではサーバ名とユーザ ID と配信識別 ID の 3 つの情報が、動画取得の際に必要なことになるという事である。

## 4 システムの実装

提案するシステムは、次のプロセスを順に実行する。

1. ユーザが URL と放送開始時刻を入力
2. at コマンドによって指定した時刻にプログラムが作動
3. サービスを提供するサイトにログイン
4. 動画の視聴に必要な引数を取得
5. 引数を適切な形に加工・RTMPDUMP に渡す

## 5 実験

実験を行った環境は次の表 1 番の通りである。

表 1 実験環境

OS	Ubuntu11.04
RTMPDUMP	v2.3
使用言語	Ruby 1.9
使用ライブラリ	Mechanize 2.0.0

### 5.1 動作確認

実際にプログラムを実行してみた所、Ustream と Stickam はすべての動画の録画保存に成功したが、ニコニコ生放送に関しては、ニワンゴ社が提供している公式放送では成功したが、一般ユーザの配信の録画予約が失敗した。RTMPDUMP に渡す引数に不具合がある可能性が高いが、原因を特定することは出来なかった。

また複数の放送を同時に録画予約した所、放送時間が被っていない場合は録画保存出来たが、放送時間が被ると RTMPDUMP が停止した。これは RTMPDUMP が同時に複数の録画予約を行うことを想定して設計されていないためと考えられる。このため現時点では複数の放送を同時に録画する事は出来ない。

## 6 おわりに

本研究はリアルタイムで配信されている動画の取得するシステムの構築と、そのシステムの拡張性を高める事を目的としていた。前者の目的は一部のサイトを除いて達

成されたが、後者に関しては達成出来ていない。RTMP サーバの URL を調べる方法として proxy サーバを経由させる方法と、パケットキャプチャしたコードから探す方法を選択したが、サイトへのアクセス記録をすべて確認し、その中から必要な情報を人間の目で探すというのは、多数のサイトの RTMP サーバの URL を調べるには効率が低い。そのため、新たに RTMP サーバの URL を自動で検索するシステムを構築する事が今後の課題である。本研究において構築したシステムと、通信内容を素早く把握する新たなシステムを一体的に運用すれば、より効率的なシステムになるだろう。

## 参考文献

- [1] Adobe: Real-Time Messaging Protocol (RTMP) specification (accessed Sep. 2012). (<http://www.adobe.com/devnet/rtmp.html/>).
- [2] Adobe Systems: AMF 3 Specification (accessed Dec. 2012). ([http://download.macromedia.com/pub/labs/amf/amf3-spec\\_121207.pdf](http://download.macromedia.com/pub/labs/amf/amf3-spec_121207.pdf)).
- [3] Andrej Stepanchuk: RTMPDump (accessed Dec. 2012). (<http://rtmpdump.mplayerhq.hu/>).
- [4] Duane Wessels: squid-cache.org (accessed Dec. 2012). (<http://www.squid-cache.org/>).
- [5] Gerald Combs: wireshark (accessed Dec. 2012). (<http://www.wireshark.org/>).
- [6] @nifty: 「事業仕分け」Ustream などネット企業 5 社が生中継 (accessed Jul. 2012). (<http://business.nifty.com/articles/topic/100422/ipdivert/>).
- [7] Ustream: Ustream Home Page (accessed Jul. 2012). (<http://www.ustream.tv/new/>).
- [8] アンテナハウス株式会社: XML 仕様の概要 (accessed Dec. 2012). (<http://www.antenna.co.jp/xml/xmllist/xml-20010920.pdf>).
- [9] ニワンゴ: ニコニコ生放送 (accessed Jul. 2012). (<http://live.nicovideo.jp/>).
- [10] 株式会社 E-Times Technologies: Stickam JAPAN! (accessed Jul. 2012). (<http://www.stickam.jp/>).