分散システム特論 レポート
RFC 1889
A Transport Protocol for Real-time Applications

情報工学専攻 荒木研究室
修士 1 年 吉村康彦

2000 年 1 月 31 日

1 はじめに

ネットワークの高速化や連続メディア処理の発達などにより、インターネット上で、動画像や音声といった実時間性の強く要求されるデータを用いた通信が行われるようになってきた。実時間性を要求するデータの通信にとって、データがある時間内に宛先へ配送されることは重要である。

しかし従来の TCP/IP アーキテクチャでは様々な階層のプロトコル間の通信や、TCP の様に信頼性を確保するために複雑な処理を行なっているため、データ紛失よりも伝送遅延や遅延揺らぎに敏感である実時間性の要求されるデータの伝送には不向きである。

そこで、実時間性が要求されるデータを伝送する際の、送信ホストと受信ホストの両端間のプロトコルを規定するものとして RTP(Real-time Transport Protocol) が提案された。

2 RTP の概要

RFC1889 はビデオ会議システムのような実時間性の要求されるデータを伝送する際の送信ホストと受信ホスト両端間のプロトコルを規定している。そして、この RFC では、実時間性のあるデータを運ぶ RTP と、QoS を監視し、RTP セッションの参加者へ情報を伝達する RTCP(Real-time Transport Controll Protocol) の仕様を記述している。

RTP では従来のプロトコルアーキテクチャに少し変更を加えている。RTP はプロトコル間の厳密なレイヤ化を緩め、レイヤ間に協調的な関係を取り入れている。このアーキテクチャは実時間アプリケーションに適して、アプリケーション層フレーミングと呼ばれている。これは TCP のような洗練されたトランスポートプロトコルではなく、アプリケーションから直接扱うことのできるような簡単な枠組みを与えるものである。

RTP はこのアーキテクチャの一部分を規定するものではあるが、完全なプロトコルを実際には規定していない。RTP はプロトコルの枠組みの概略を示したものである。具体的なアプリケーションはそのアプリケーションに必要なもの (ペイロードフォーマットの規定) をこの枠組みに付け加えることによって構築され、組み合わされたものが完全なプロトコルとなる。

また、RTP は下位層のプロトコルには依存しないので適切なプロトコルを使うことでより効果を発揮できる。例えば下位層のプロトコルがマルチキャストの機能を提供できれば、複数の送付先ホストへ同時にデータを配信できる。

RTP それ自体は QoS(Quality of Service) を保証する機構を備えているわけではなく、それも下位層における機能に依存する。

3 RTP メッセージ

RTP メッセージのフォーマットは図 1 のようになる。

RTP メッセージは、全て同一のフォーマットを使用する。メッセージフォーマットから分かるように RTP は非常に簡潔ななヘッダを使用する。これは、RTP がアプリケーションデータに大きな負荷を加えないようにするためである。実時間アプリケーションは大きな帯域幅を使用する傾向があり、またその多くは遅延を小さくするためにサイズの小さなメッセージを使用するので、ヘッダが小さいことは効率的な転送に有効である。

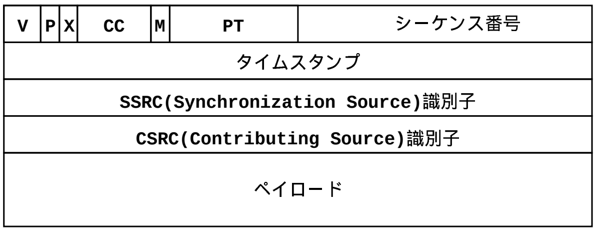


図 1: RTP メッセージフォーマット

また、RTP はアプリケーション層フレーミングを支援するので、このメッセージフォーマットは個々のアプリケーションが必要とする様々な解釈や追加を行うのに適している。

以下に RTP メッセージの各フィールドについて説明する。

V(Version):2bits
: RTP のバージョンを識別する。RFC1889 の段階では 2 である。

P(Padding):1bits
: そのパケットにパッドを詰めるための余分なバイトが付加されているかどうかを示す。

X(Extension):1bits
: メッセージが基本 RTP ヘッダの後に拡張ヘッダを持つ場合にセットされる。

CC(CSRC Count):4bits
: そのメッセージがいくつの貢献送信元識別子 (CSRC) を含むかを示す。このフィールドのサイズは 4 ビットなので貢献送信元の数が 15 を越えた場合、15 個だけが明示的に示される。

M(Marker):1bits
: アプリケーションが指定可能なビット。一般に、アプリケーションはデータの境界に印をつけるためにこのフィールドを使用する。

PT(Payload Type):7bits
: メッセージのペイロードタイプを指定する。ペイロードタイプとは符号化形式のことである。

シーケンス番号:16bits
: パケットの送信順に振られる自然数である。

タイムスタンプ:32bits
: そのパケットの先頭のバイトが送信された時刻を示す値を含む。

SSRC(Synchronization Source) 識別子:32bits
: メッセージの元々の送信元を識別するために使用される。同期送信元とは、そのデータに対するシーケンス番号と時刻印を指定したホストのことである。

CSRC(Contributing Source) 識別子:32bits
: メッセージに含まれる各ストリームを準備したホストを識別するために使用される。

4 RTCP メッセージ

RTP はアプリケーションデータを運ぶメッセージというただ 1 つのメッセージタイプしか規定していない。しかし、多くの実時間アプリケーションではフィードバックや応答メッセージが必要となる。そこで RTP はこれらの機能を別のプロトコルである RTCP(Real-time Transport Protocol) に委ねる。

RTP トラフィックと RTCP とは密接に関係しているため、使用するポートは、RTP が偶数のポートを使い、RTCP はそれより 1 大きいポートを使用する。

また、RTCP ではパケットタイプ値によって識別できる 5 つのメッセージタイプがある。以下でその各々のメッセージタイプについて説明する。尚、( ) 内の値はそのメッセージのパケットタイプ値である。

送信者報告 (200)
: データ送信者が、ホストの送信 (受信) 状態をデータ送信先や、第三者のモニタに通知する。NTP (Network Time Protocol) と同じフォーマットの絶対時間と、それに対応する RTP タイムスタンプ、送信した RTP パケット数、RTP バイト数、複数の受信者ブロックを送付する。受信者ブロックには必要に応じて受信者報告と同じものを挿入する。

受信者報告 (201)
: 受信した RTP データの取得状況について、個々のデータ送信元や第三者のモニタに通知する。複数の送信元それぞれに、一つのブロックを使用し、前回の報告を発行した後から積算した送信元からのパケット紛失率と、紛失パケット数、受信した最大シーケンス番号、到着間隔 (ジッタ)、送信元から

の最後に受け取った送信者報告パケットの時刻と、その時刻からの経過時間を挿入する。

送信元記述 (202)
: 送信元が自分についての情報を受信者や第三者のモニタに通知する。送信元の持つ一意でで曖昧性のない識別名 (CNAME)、実ユーザ名 (NAME)、電子メールアドレス (EMAIL)、電話番号 (PHONE)、住所 (LOC)、ストリーム生成ツール (TOOL)、注釈 (NOTE)、独自拡張項目 (PRIV) などを挿入する。

退去メッセージ (203)
: 送信元がそのセッションから退去することをネットワーク上へ通知する。退去理由を示すオプションを挿入することができる。

アプリケーション独自のメッセージ (204)
: アプリケーション独自の利用法のために確保している拡張である。これは主に、アプリケーションが新しい種類のメッセージでの実験を行うための簡易な方法を提供する。

5　RTP の伝送方式

図 に RTP 及び RTCP の伝送方式の一例を示す。

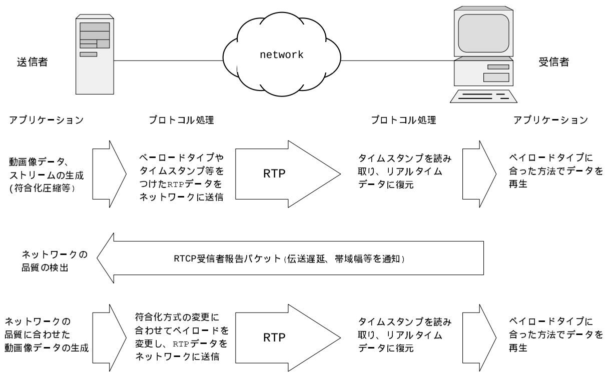


図 2: RTP の伝送方式例

メディアサーバからネットワークを介して動画像、音声を再生する場合、サーバ側はアプリケーションから受け取ったストリームデータにペイロードタイプ、タイムスタンプなどの情報を付け RTP パケットとして送出する。クライアントは受信したデータを RTP ヘッダよりストリームデータとして復元し、アプリケーションに渡す。

また、定期的に RTCP パケットがサーバ・クライアント間で伝送され、RTCP は回線品質の状態の検出及び通知のために利用される。アプリケーションは通知された回線品質に合わせたビットレートでストリームデータを生成する。

6　トランスレータとミキサ

RTP でのストリームデータの送信において、送信者と受信者というエンドシステムの役割の他にトランスレータとミキサという二つの中継システムの役割が定義されている。この二つは、ネットワークを通過する RTP データにネットワークの能力に応じた加工を施して下流へデータを受け渡す。

トランスレータとは、あるペイロードフォーマットから別のフォーマットへ「翻訳」する機能である。例えば能力の低い端末や、帯域の狭いネットワークへ向けてデータを伝送する際にペイロードを変換してデータ量を削減するする役割を果たす。トランスレータにおいてはペーロードを加工するだけで、RTP ヘッダに含まれる SSRC（同期送信元）識別子、CSRC（貢献送信元）識別子の変更は行わない。

一方ミキサは、複数の送信元からのデータストリームをフォーマットの変換をすることなく、1 つのストリームに組み合わせる機能を持っている。この機能は複数の参加者が存在する音声会議などで有効である。ミキサにおいてはメディア間の再同期を行うため、SSRC 識別子はミキサ自身を示す値に書き換えられ、ペイロードの要素となる送信元は CSRC 識別子として追加する。

7　RTP の実装例：vic

現在、RTP を実装している様々なアプリケーションが存在するが、本発表では RTP の実装例として vic(Video Conferencing Tool) を紹介した。

vic は UCL(University of California, Berkeley)/LBNL(Lawrence Berkeley National Laboratory) で開発されたインターネット上でのビデオ会議用のアプリケーションであり、映像の送受信を行うことができ、IP マルチキャストにも対応している。図 3 に vic のユーザインターフェイスを示す。

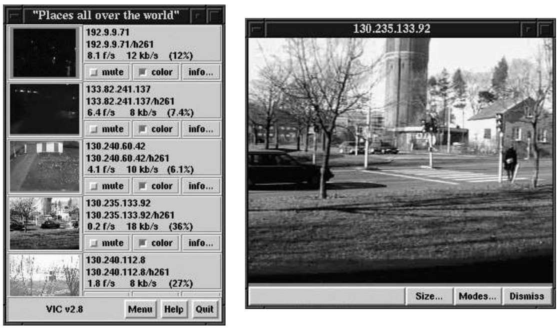


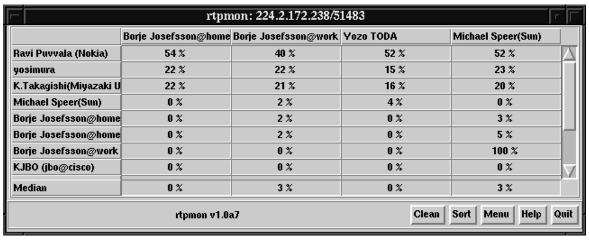
図 3: vic のユーザインターフェイス

7.1 RTCP メッセージの解析

vic を用いて特定のマルチキャストグループに参加し、その際流れる RTCP メッセージの解析を rtpmonitor というツールを用いて行った。

図 4 は RTCP メッセージから各受信者のパケット紛失率を表示したものである。RTP 及び RTCP は下位層のプロトコルには依存しないため、RTCP メッセージもマルチキャストで送信される。そのため他の受信者の受信状況を把握することも可能である。

また、図 5 は RTCP の送信元記述メッセージを表示したものである。



rtpmon: 224.2.172.238/51483

	Borje Josefsson@home	Borje Josefsson@work	Yozo TODA	Michael Speer(Sun)
Ravi Puvvala (Nokia)	54 %	40 %	52 %	52 %
yosimura	22 %	22 %	15 %	23 %
K.Takagishi(Miyazaki U	22 %	21 %	16 %	20 %
Michael Speer(Sun)	0 %	2 %	4 %	0 %
Borje Josefsson@home	0 %	2 %	0 %	3 %
Borje Josefsson@home	0 %	2 %	0 %	5 %
Borje Josefsson@work	0 %	0 %	0 %	100 %
KJBO (jbo@cisco)	0 %	0 %	0 %	0 %
Median	0 %	3 %	0 %	3 %

rtpmon v1.0a7 Clean Sort Menu Help Quit

図 4: 受信者のパケット紛失率

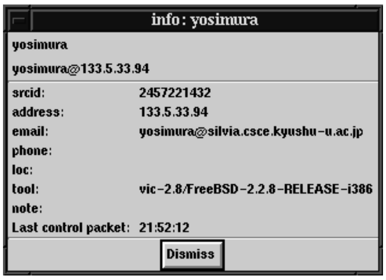


図 5: 送信元記述

8 まとめ

本発表では RTP とそれに付随する RTCP について説明し、その実装例として vic の紹介を行った。

RTP と RTCP は基本的には音声会議やビデオ会議のために設計されているが、分散シュミレーション・モデリングや、実時間制御など多くの応用が可能である。

また、RTP と RTCP はマルチキャスト指向性をもっている。この設計は基本として考えている環境である音声会議及びビデオ会議の特徴を反映している。これらのアプリケーションではデータメッセージはもちろん、フィードバックメッセージや応答メッセージ配送にもマルチキャストを使用する。