

分散マルチメディアシステムにおける QoS 機能テストの一手法*

5L-06

孫 太 安本 慶一 森 将豪†
滋賀大学経済学部情報管理学科‡

1 はじめに

近年、マルチメディアシステムの実現における正しさをテストする方法が注目されている。マルチメディア通信システムのテストに関しては、伝送系 [3] やコンテンツの品質 [4]、分散システムの相互操作性やパフォーマンステストに関する研究 [6]、等があるが、いずれも時間関係のテストは含まれていない。最近になって、メディアオブジェクト間の時間関係を対象としたテスト方法に関する研究として [2, 5] が報告されている。

本稿では、分散マルチメディアシステムにおける QoS をテストする方法を提案する。一般に、マルチメディアシステムでは、動画や音声などのメディアオブジェクトが指定した品質 (フレームレートや画質、音質など) で再生できているかどうかや、エンコード時のタイムスケール通りに再生できているかどうか、複数のメディアオブジェクト間のが指定した時間範囲に収まるよう制御されているかどうか (メディア同期)、などが機能テストの対象となる。本稿では各メディアオブジェクトの再生動作の仕様を時間オートマトンで与え、仕様間で成り立つべき依存関係を、時間制約付き時相論理の一つである TCTL を用いて指定し、実装 (IUT) がこれらの仕様に指定された通りの QoS を満たすかどうかをテストする提案手法について述べる。

2 対象システムと機能テストの概要

2.1 マルチメディアシステムの仕様記述

オブジェクト単体の再生動作の仕様 本稿では、8 項組 $M = \langle S, A, C, V, Guard, Def, \delta, s_0 \rangle$ で定義される拡張タイムオートマトンによりシステムの動作を記述する。ここで、 $S = \{s_0, s_1, \dots, s_n\}$ は状態の有限集合、 A はアクション (イベント) の有限集合、 C はクロックの有限集合である。また、 V は変数の集合、 $Guard$ はアクションの実行条件の集合、 Def は変数への値の代入文 (クロックのリセットを含む) の集合、 $\delta: S \times A \times C \rightarrow S \times V$ は状態遷移関数、 s_0 は初期状態である。各イベントはゲートでの値の入出力であり、 $a?x, b!f(y)$ のように表記する。

本稿では、図 1 のようなマルチメディアシステムを対象とする。クライアントシステムはゲート u, n, v を持ち、それぞれ、ユーザ、ネットワーク、画面とのインタフェースとして使用するとする。クライアントシステムは指定した品質値のマルチメディアオブジェクトに対するストリームをサーバから受信し再生する。品質値 q は、 $1 \leq q \leq Q_{max}$ の範囲で定義されており、大きいほど良い品質を表すものとする (ただし、ここでは品質として時間解像度のみを考える)。簡単のため、サーバからのストリームは、各単位データ (画像フレームなど) 毎に一つのパケットとしてクライアントシステムに受信されるものと仮定する。ただし、パケットは消失するかも知れない。また、パケットの受信状況が一定期間良くない場合、あるいは良好な場合を、それぞれ、イベント $congestion, stable$ により知ることができるとする。

以上の仮定のもとでのクライアントシステムの動作の記述例を図 2 に示す (図の 2 重丸は初期状態)。仕様では、要求が受信品質値 q に対し、フレーム 1 個を再生すべき時間周期を $P(q)$ とすると、受信 ($n?frame$)、デコード ($rawdata := decode(frame)$)、再生 ($vravdata$) といった一連のイベント系列は $P(q)$ 以内に実行されねばならない。仕様では、リアルタイム性の確保のため、 $P(q)$ 時間経過した場合には、上記の全てあるいは一部をスキップするよう記述している (図中の点線の遷移)。一方、パケットの受信状況が悪い場合 ($congestion$)、あるいは良好な場合 ($stable$) には、品質値 q を増減させ、サーバに新しい品質値のストリームの配信を要求するよう記述している。

複数オブジェクト間の依存関係の記述 オブジェクト間の依存関係は TCTL [1] で記述する。例えば、図 2 で示され

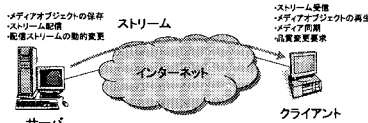


図 1: 対象とする分散マルチメディアシステム

るビデオ再生と音声再生タイムオートマトンを並行して実行することを指定するとする。ここで、フレームが出力されるゲートをそれぞれ v, a 、ゲート v において先頭から c 個目のビデオフレームが出力されたことを述語 $FOut(v, c)$ で表すものとする。最新のビデオフレーム、オーディオフレームの通し番号を c_v, c_a で表す。ビデオ、オーディオの処理時間単位を P_v, P_a と表記する。ビデオと対応するオーディオのずれが常に 80ms 以内であるという性質は、TCTL を用いて以下のように記述できる。

$$\square((FOut(v, c_v) \rightarrow \diamond(FOut(a, c_a) \wedge c_a \cdot P_a - c_v \cdot P_v \leq 80)) \wedge (FOut(a, c_a) \rightarrow \diamond(FOut(v, c_v) \wedge c_v \cdot P_v - c_a \cdot P_a \leq 80))) \quad (1)$$

さらに、理想的な (本来の) 再生時刻より 1000ms 以上ずれることはないという性質についても以下のように記述できる (変数 $clock$ は再生開始時点からの経過時刻)。

$$\square(clock \geq c_v \cdot P_v \rightarrow \diamond \leq 1000 FOut(v, c_v) \wedge clock \geq c_a \cdot P_a \rightarrow \diamond \leq 1000 FOut(a, c_a)) \quad (2)$$

2.2 機能テストの概要

本稿では、以下の 2 つのテストを実現する。

2.2.1 各メディアオブジェクトの再生品質に対するテスト

本稿では、入力ストリームにおけるジッタおよびパケットロス、およびクライアントシステムへの負荷を考え、オブジェクトが再生されるべき品質をこれらの値の関数として定義し、与えられたストリームおよびシステムの負荷に対し、オブジェクトの各単位データ (動画フレームや音声の断片など) の出力時刻を観察することにより、定義された品質を満たしているかどうかをテストする。

本稿では、再生されるべき品質 fps' を次の様に定義する。

$$fps' = fps \cdot (1 - \alpha \cdot LossRatio) - \beta$$

ここで、 fps は元データ (サーバが送信するストリーム) のフレームレート、 $LossRatio$ はパケットロス率、 α は、各パケットの消失が画像フレームの消失に関与する割合である (例えば、モーション JPEG 動画では $\alpha = 1$ となり、一つの画像フレームが複数パケットで送信される場合や、MPEG などフレーム間の依存関係がある場合、 α の値は大きくなる)。 β はクライアントシステムにおけるネットワーク以外の変動要因 (CPU 負荷など) である。

品質関数に基づいた品質および輻輳検出機構が正しく実装できているかは、(1) 一連のパケットを入力として IUT に与え (パケットの入力時刻は、各品質値 q のオブジェクト配信を満たす範囲でランダムに与える)、(2) ある時間周期 MP の間、パケットロス率およびフレームレートを計測し、

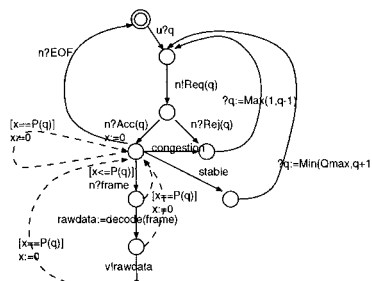


図 2: オブジェクト再生動作の仕様例

* A Method for QoS Functional Testing in Distributed Multi-media Systems

† Tao Sun, Keichi Yasumoto, Masaaki Mori

‡ Dept. Info. Proc. and Man., Shiga University

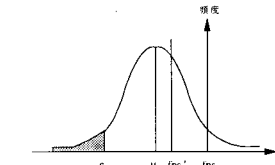


図 3: フレームレートの正規分布

上記の関係を満たすかどうか調べることでテストする。また、上記(1),(2)に加え、パケットロス率の増減に伴う品質値の切り替えがクライアント・サーバ間でうまく動作しているかどうかをテストする。

2.2.2 複数オブジェクトの依存関係を考慮した機能テスト

あるクライアントが複数のメディアオブジェクトを受信している場合、オブジェクト間には、動画と対応する音声、字幕などの間のメディア同期、ビデオ会議などにおける、オブジェクト間の優先度などの依存関係がある場合が多い。

このような依存関係のある複数並行オブジェクトをテストする際には、複数の対応するタイムアウトの直積マシンを構成し、それに対しテスト系列を求める方法があるが、状態爆発およびテスト系列の肥大化などの問題が生じる。本稿では、各オブジェクトの再生品質を単体でテストするテスト系列と、オブジェクト間の依存関係のためのテスト系列を別々に生成し、それらをテストで同期並列実行することによりテストを行う。

3 テスト系列の生成

3.1 単一オブジェクトに対するテスト系列の生成

マルチメディアオブジェクトの再生機能テストは、プロトコル適合性試験と異なり、意味のある動作系列の任意回の繰り返しを含む一連の生起事象系列を対象として考察せねばならず、テストの成否は所与の時間幅の中で総合的に判定されるべきものである。したがってテスト系列は、与えられたクライアントの仕様記述より繰り返し実行される動作系列を抽出し、それらの正規表現として与えるのがよく自然であると考えられる。

中間テスト系列の導出 与えられた仕様記述より一連の意味のある独立した繰り返し動作系列に着目し、それらを部分系列として表現する。そして、得られた部分系列から仕様が実現する生起事象系列全体を求める。動画送信のフレームレートのテストを対象とした図2の仕様例では、初期状態でクライアントが初期品質要求がなされる部分系列を T_0 、品質要求が叶えられず品質低下を行う部分系列を T_1 、ある品質の下でオブジェクト再生が行われる部分系列を T_2 、および、パケット損失の観測結果から外部モニタ機構により送信品質を変更するための部分系列を T_k とすれば、それぞれ以下のように表される：

$T_0 : ulq\{GC := 0\}$

$T_1 : n?Req(q).n!Rej(q)$

$T_2 : \{clock := 0\}.n!Acc(q).(n!frame[0 \leq clock \leq P(q)]\{pn := pn + 1\}.\{clock \geq P(q)\}\{clock := 0\} + v?rawdata[clock \leq P(q)]\{vn := vn + 1\}.\{clock \geq P(q)\}\{clock := 0\}\}^* + [clock \leq P(q)]\{clock := 0, ln := ln + 1\}$

$T_k : [GC \geq MP].\{vn/MP = (1 - (ln/(pn + ln))) \cdot fps\}.\{GC := 0, pn := 0, vn := 0, ln := 0\}$

したがって、テストの対象となる系列の全体は $T_0.(T_1 + (T_2^* + T_k))^*$ で表される。

具体的なテスト系列 演算 + と * に対して選択確率と繰り返し回数をそれぞれ $+(0.1, 0.9)$ や $*(30)$ のように陽に指定することで具体的なテスト系列を定める。例えば、 $T_1 + (0.1, 0.9) T_2^{(30)}$ と指定された場合には、系列 T_1 と系列 T_2 はそれぞれ 10% と 90% の確率で実行し、系列 T_2 の実行に際しては繰り返し 30 回行うことが指定されており、テストがこれを実現して IUT に与えることになる。テストは IUT に対するフレーム出力タイミングをも制御する。フレームレートのテスト評価方法 十分に長いモニタ周期 (MP) 間を通じて観測・計算されるパケットロス率は、MP よりもずっと短い周期 (例えば 1 秒程度) における再生においては常に変化するという意味において、マルチメディア・オブジェクトシステムは非決定的である。それゆえに、モニタ周期で観測されるタイミングテストの出力には曖昧さが生じる。これに対応するために、クライアント側で実現されるフレームレートの分布を仮定し、IUT からのごく

短い周期でのテスト出力を集約したデータの分布をもとに統計的に処理を行い、ある信頼性をもってテストにパスしたか否かを判定する方法を採用する [2]。具体的には、テストは、モニタ周期 (MP) と送信品質 (パケット伝送速度) を設定し、設定された周期的なパケット送信時刻にランダムなジッタ成分を加えた時刻にフレームを IUT に与え、設定しておいたモニタ周期ごとに IUT におけるパケットロス率を計測し、2.2.1 節で定義した式より当該周期間の平均再生フレームレート fps' を求める。一方、テストは 1 秒ごとに IUT からの出力を観測し、多くの繰り返しテスト試行よりモニタ周期にわたる 1 秒ごとのフレームレートの値の集合を標本として確保する。この標本空間よりフレームレートの平均と標準偏差を求め、テスト側で設定しておいた ϵ (IUT に対する最大許容度) より決まる図3の網掛部分の面積と、予め決めておいたリジクチュールされる信頼レベルとの比較によりテスト結果を判定する。ここでは簡単のため、フレームレートの分布は正規分布に従うものと仮定している。

3.2 複数オブジェクトに対するテスト系列の生成

2.1 節で述べた、ビデオと音声の間に TCTL の式 (1),(2) によってメディア同期が指定されたシステムに対し、ビデオと対応する音声オブジェクトに対する単体テスト系列 T_{qv}, T_{qa} には、フレームの再生イベント v, a が繰り返し現れる。これらのイベントが式 (1),(2) の性質を満たすには、ゲート v, a のイベントがこれらの性質を満たす場合のみ実行可能なようにガード式を加えねばよい。すなわち、対応するテスト系列 T_s は次のようになる。

$T_s : (nv?Acc(q1).na?Acc(q2) + na?Acc(q2).nv?Acc(q1))^*.\{clock := 0, cv := 0, ca := 0, P_v := P(q1), P_a := P(q2)\}.\{v?pic?c_v \cdot P_v - c_a \cdot P_a \leq 80 \wedge clock \geq c_v \cdot P_v\} + a?wav?c_a [c_a \cdot P_a - c_v \cdot P_v \leq 80 \wedge clock \geq c_v \cdot P_v] + v!EOF + a!EOF)$

ここで、 nv, na は、ビデオ、音声オブジェクト再生プロセスのネットワークとのインタフェースとなるゲートである。求めた T_s を T_{qv}, T_{qa} とゲート n, v, a に関して同期並列実行することでテストを実現する。

4 テスタの構成

単一オブジェクトに対するテスト系列には、各分岐の選択確率と繰り返し回数が指定されているので、テストは、これらの値に基づき、各分岐を乱数発生により選択し、繰り返しは指定回数を反復実行する。本テストは系列内の各出力イベントの実行時刻を指定範囲内でランダムに選ぶ。例えば系列 $a! [clock < 4] b? [3 < clock < 9]$ の場合、0 から 4 までの範囲内でランダムな時刻を選んでイベント a を実行し、その出力を IUT に与える。入力イベント $b? [3 < clock < 9]$ については、IUT からの入力が指定範囲 $3 < clock < 9$ にあるかどうかを調べれば良い。

複数オブジェクトに対するテストの場合は、対応する複数のテスト系列と依存関係のチェックのためのテスト系列を、並列同期実行しなければならない。本研究では、Java 言語のマルチスレッド機構とその間の共有変数を用いて、複数テスト系列群の並列処理およびそれらの間の同期を実現する。

5 おわりに

本稿では、分散マルチメディアシステムにおける QoS に関する機能テストの一手法を提案した。今後の課題として、フレームレート以外の再生品質についてのテスト手法および、メディア同期以外 (優先順位など) の依存関係を持った複数並行オブジェクトのテスト手法の考案が挙げられる。

参考文献

- [1] Alur, R. and Henzinger, T. A.: "Logics and Models of Real Time: A Survey", "Real Time: Theory in Practice", LNCS 600, pp.74-106 (1992).
- [2] Cheung, S.C., Chanson, S.T. and Xu, Z.: "Toward Generic Timing Tests for Distributed Multimedia Software Systems", IEEE Int. Sympo. on Software Reliability Engineering (2001).
- [3] Fibush, D.K.: "Testing multimedia transmission systems", IEEE Design & Test of Computers, Vol.12, No.4, pp.24-44 (1995).
- [4] Grabowski, J. and Walter, T.: "Testing Quality-of-Service Aspects in Multimedia Applications", Proc. of 2nd Workshop on Protocols for Multi-media Systems (PROMS) (1995).
- [5] Yamada, M., Mori, T., Fukada, A., Nakata, A. and Higashino, T.: "A Method for Functional Testing of Media Synchronization Protocols", to appear in Proc. of the 16th Intl. Conf. on Information Networking (ICOIN-16) (2002).
- [6] Walter, T., Scheferdecker, I. and Grabowski, J.: "Test Architectures for Distributed Systems - State of the Art and Beyond (Invited Paper)", Testing of Communication Systems, Vol.II, Chapman & Hall (1998).