

やわらかいビデオ会議システムにおける リスク値を用いた QoS 調整戦略導出手法

武田 敦志[†] 今井 信太郎^{††} 北形 元^{††}
菅沼 拓夫^{††} 木下 哲男^{†††} 白鳥 則郎^{††}

本論文では、やわらかいビデオ会議システムにおいて、QoS 調整の場面にに対して効果的である QoS 調整戦略を、効率良く導出する手法について提案する。効果的 QoS 調整戦略導出の計算処理量を削減するためには、QoS 調整戦略の検証回数を減らす必要があるが、検証回数を減らすと導出の正確性が損なわれる可能性がある。本論文では、QoS 調整戦略の動作履歴からリスク値を算出し、このリスク値を用いて QoS 調整戦略の検証順序を決定することにより、少ない計算処理量で効果的な QoS 調整戦略を正確に導出する。また、提案手法を適用したプロトタイプシステムを用いて実験を行い、従来手法と比較評価することにより、提案手法の有効性を検証する。

The Method to Derive a QoS Control Strategy by Using Risk-value on Flexible Videoconference System

ATUSHI TAKEDA,[†] SHINTARO IMAI,^{††} GEN KITAGATA,^{††}
TAKUO SUGANUMA,^{††} TETSUO KINOSHITA^{†††} and NORIO SHIRATORI^{††}

In this paper, we propose an efficient method to derive a QoS-control-strategy which is effective for the QoS-control situation in Flexible Videoconference System. In order to decrease the amount of calculation to derive an effective QoS-control-strategy, it is required to decrease a number of QoS-control-strategy verifications. However, decreasing a number of verifications can make accuracy of derivation ruined. Therefore, in this paper, we propose the method to derive an effective QoS-control-strategy accurately with a small amount of calculation. The method is realized by using risk-value which is calculated from the behavior record of QoS-control-strategy. We show the effectiveness of our proposal from the experimental results that is provided by prototype system.

1. ま え が き

インターネットをはじめとするコンピュータネットワークの充実や計算機端末の高性能化により、一般家庭の PC やモバイル端末などの汎用 PC と ADSL や無線によって接続されたインターネット環境において、IP 電話や多人数参加型ゲームなど、動画や音声などのデータを送受信するマルチメディア通信システム

(MCS) が普及しつつある。このような環境においては、MCS 利用と同時に、文書整形や WEB の閲覧などの、MCS と共通の計算機資源を使用するアプリケーションを利用するのが一般的である。MCS はリアルタイム性が重要なアプリケーションであるため、そのプロセス優先度やパケット優先度は他のアプリケーションより高く設定される。しかし、MCS が CPU やネットワーク帯域などの計算機資源の大部分を使用すると、同時に動作しているアプリケーションが必要な計算機資源を確保できず、利用者要求を満足させる動作を実行できないという問題がある。そのため MCS には、他のアプリケーションの動作を観察し、刻々と変化する計算機資源の状態に適応して、MCS が提供する動画や音声などの QoS と MCS が利用する計算機資源を調整する機能が求められている。

計算機資源の状態の変化に適応して QoS を調整する MCS として、やわらかいビデオ会議システム (FVCS)

[†] 東北文化学園大学科学技術学部応用情報工学科
Department of Applied Information Technology, Tohoku Bunka Gakuen University

^{††} 東北大学大学院情報科学研究科/電気通信研究所
Graduate School of Information Sciences, TOHOKU University/Research Institute of Electrical Communication

^{†††} 東北大学大学院情報科学研究科/情報シナジーセンター
Graduate School of Information Sciences, TOHOKU University/The Information Synergy Center

などが提案されている¹⁾。FVCS は計算機資源の状態変化に適応した QoS の調整 (QoS 調整) を行う際に、QoS 調整の指針 (QoS 調整戦略) を決定し、その戦略に従って QoS 調整を実行する。QoS 調整戦略には、大胆な QoS 調整を行う戦略や慎重に QoS 調整を行う戦略などがあり、FVCS はこれらの戦略を適宜切り替えることによって種々の状況に適応する。ただし、QoS 調整が必要な状況に対して効果的な QoS 調整戦略を実行するためには、実行する QoS 調整戦略を決定するための知識 (QoS 調整戦略決定知識) に、MCS が利用している計算機資源の利用状態 (計算機資源状態) の変化特性を反映させる必要がある。

我々は、MCS が計算機資源状態の変化特性についての情報を収集し、それらを自動的に解析して知識とすることにより、計算機資源状態の変化特性を反映させた QoS 調整戦略決定知識を自動的に獲得する手法の提案を行ってきた²⁾⁻⁴⁾。この技術を IP 電話や多人数参加型ゲームなどの MCS に適用することにより、これらの MCS において的確な QoS 調整が実現できると考えられる。しかし、この知識を獲得するためには、それぞれの状況に対して効果的な QoS 調整戦略 (効果的 QoS 調整戦略) を導出する計算が必要である。従来手法では、MCS に登録されている QoS 調整戦略の増加にともない、効果的 QoS 調整戦略を導出するための計算処理量も増加するという問題がある。この計算処理量が増加すると、MCS のサービスを提供するための計算機資源が圧迫されるため、MCS が提供する QoS が低下する。MCS がより多くの環境に適応するためには、それらの環境で効果的に働く QoS 調整戦略を追加する必要があるため、端末の計算能力やネットワークへの接続環境などが異なる多様な環境に適応する MCS を実用化するうえで、この計算処理量の増加は重大な問題である。

そこで本論文では、MCS の動作履歴に基づき、採用された QoS 調整戦略を評価し、その評価を活用することにより、それぞれの状況で効果的な QoS 調整戦略を、従来より少ない計算処理量で導出する手法を提案する。提案手法では、最も効果的な QoS 調整戦略を導出する際に、それぞれの QoS 調整戦略に対して行う、QoS 調整のシミュレート (QoS 調整テスト) とその結果の評価を途中で打ち切ることにより、従来よりも計算処理量を減少させる。しかし、QoS 調整のテストを実行する順序によっては、QoS 調整戦略決定知識が不正確になり、QoS 調整を行うときに効果的な QoS 調整戦略を採用することができず、QoS 調整の的確さが低下する場合がある。提案手法では、過去

の QoS 調整戦略の実行頻度だけではなく、QoS 調整戦略の動作特性も考慮して QoS 調整テストの実行順序を決定する。この手法により、従来よりも少ない計算処理量で効果的な QoS 調整戦略を導出しながらも、従来と同等に的確な QoS 調整を実現している。

本論文では、効果的 QoS 調整戦略を効率良く導出する手法を提案する。さらに、プロトタイプシステムを用いた実験を通じ、効果的 QoS 調整戦略を導出するための計算処理量が従来より減少し、かつ、QoS 調整の的確さは従来と同等であることを示す。

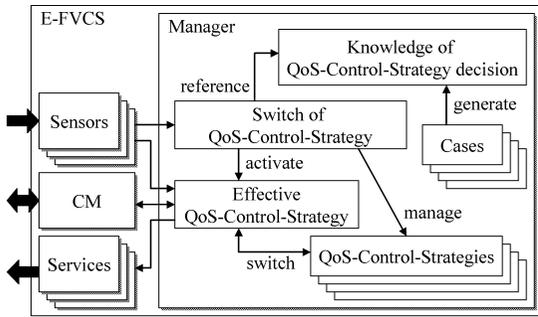
以下 2 章で QoS 調整戦略決定知識を自動的に獲得する MCS である E-FVCS の動作を示し、3 章で効果的 QoS 調整戦略を効率良く導出する手法について提案する。また、4 章では提案手法の有効性を実験結果から検証し、5 章では提案手法の適用範囲と応用について検討する。最後に、6 章で本論文のまとめと今後の課題について述べる。

2. E-FVCS: Effective Flexible Video-Conference System

2.1 E-FVCS の概要

QoS 調整の指針となる戦略 (QoS 調整戦略) を切り替えることで、ネットワークの輻輳や CPU 資源の枯渇など、計算機資源の利用状態 (計算機資源状態) の様々な変化に適応するマルチメディア通信システム (MCS) として Flexible VideoConference System (FVCS) がある¹⁾。FVCS における QoS 調整戦略の実行方法は、最初に慎重な戦略を実行し、その戦略による QoS 調整が時間を要するならば、徐々に大胆な戦略に切り替えるという単純なものであった。これに対して我々は、「計算機資源に対して同時に影響を与えるアプリケーションは少数である」という仮定のもとで動作環境の情報を収集・解析することにより、QoS 調整を行う状況に対して効果的な QoS 調整戦略 (効果的 QoS 調整戦略) を決定する知識 (QoS 調整戦略決定知識) を自動的に獲得する手法を提案してきた^{2),3)}。この知識獲得手法を適用した MCS である Effective Flexible VideoConference System (E-FVCS) では、計算機資源状態の変化特性に適応した QoS 調整戦略決定知識を獲得し、その知識を用いて QoS 調整の状況に対して効果的 QoS 調整戦略を導出することにより、的確な QoS 調整を実現している。

図 1 に E-FVCS の構成の概要を示す。E-FVCS では、動作環境の情報や E-FVCS の動作履歴を事例 (Case) として蓄え、それを解析することにより QoS 調整戦略決定知識 (Knowledge of QoS-control-strategy



CM : Communication Manager to negotiate QoS with the other side
 Services : Media Controller (Video, Sound, Text and etc)
 Sensors : Resource Sensor (CPU-usage, Network-bandwidth and etc)

図 1 E-FVCS の構成の概要

Fig. 1 Conceptual architecture of E-FVCS.

decision) を獲得する。QoS 調整を行うときには、この知識を用いることにより、E-FVCS に登録されている QoS 調整戦略 (QoS-control-strategy) の中から効果的 QoS 調整戦略 (Effective QoS-control-strategy) を決定する。

2.2 関連研究

QoS 調整を行う MCS に関しては従来から多くの研究報告がなされており、計算機資源を確保することで安定した QoS を提供することを目標とする研究⁵⁾⁻⁷⁾ や、限られた計算機資源に適応した QoS を提供することで利用者要求をできる限り満足させることを目標とする研究^{8),9)} がある。

計算機資源を確保し安定した QoS を提供することを目指した研究として、MCS に CPU 資源を割り当てる研究⁶⁾ や、MCS を含めた複数のアプリケーションにネットワーク帯域を割り当てる研究⁷⁾ がある。これらの研究では計算機資源を MCS に優先的に割り当てることにより QoS を保証することを目指している。しかし、モバイル端末や無線ネットワークなどの計算機資源に余裕のない環境の場合、MCS がすべての計算機資源を占有し、MCS 以外のアプリケーションが十分に動作しないという問題が発生する。対して E-FVCS では、E-FVCS 以外のアプリケーションが計算機資源を必要とするときは、E-FVCS が提供する QoS を低下させ、E-FVCS が使用している計算機資源を解放することにより、他のアプリケーションが十分に動作できるだけの余剰計算機資源を確保する。

限られた計算機資源に適応した QoS を提供することで利用者要求をできる限り満足させることを目指した研究としては、使用する計算機資源と利用者に提供する QoS の関係をファジークラシファイアシステムを用いて獲得する研究⁸⁾ や、MCS が提供する QoS と利

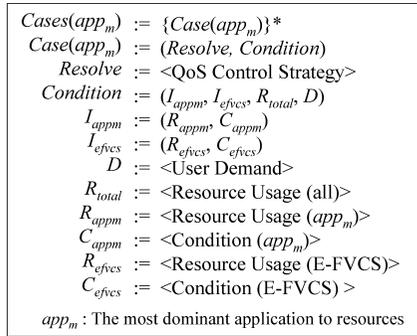


図 2 QoS 調整戦略決定知識獲得のための事例に含む情報
 Fig. 2 Information in a case to generate knowledge to derive QoS-control-strategy.

用者が感じる QoS との関係機械学習を通じて獲得する研究⁹⁾ がある。これらの研究では、計算機資源状態から MCS が提供する QoS を決定することを実現している。しかし、計算機資源状態の変化に対応するための戦略は 1 つに限られるため、計算機資源状態の変化特性によっては、MCS が提供する QoS が不安定となる問題や、計算機資源の変化に適応するまでの時間が長くなるという問題が発生する。対して E-FVCS では、その状況に対して効果的な QoS 調整戦略を選択し、その戦略に従って QoS 調整を行うため、従来よりも多くの状況での確な QoS 調整が可能となる。

2.3 QoS 調整戦略決定知識の獲得

図 2 に E-FVCS が QoS 調整戦略決定知識を獲得するために収集する事例に含まれる情報を示す。E-FVCS では、動作環境についての情報のうち、QoS 調整戦略の決定に関係する情報のみを事例として収集し、それらを解析してルールを生成することにより QoS 調整戦略決定知識を獲得する。事例は条件と結論の組になっており、条件は E-FVCS が QoS 調整を実行したときの状況であり、結論はその状況において効果的であった QoS 調整戦略である。図 2 におけるアプリケーション app_m は、計算機資源の使用量が最も変化したアプリケーションや計算機資源の使用量が最も多かったアプリケーションなどの、そのとき計算機資源の状態に最も影響を与えたアプリケーションである。アプリケーション app_m の情報を詳しく収集することにより、効率の良い QoS 調整戦略決定知識の獲得を実現する。また、アプリケーション app_m が異なると、図 2 の事例に含まれる属性値の意味も異なるため、この事例はアプリケーション app_m ごとに分類し蓄積される。

図 3 に E-FVCS の QoS 調整と事例収集の動作手順を示し、以下にその手順について述べる。

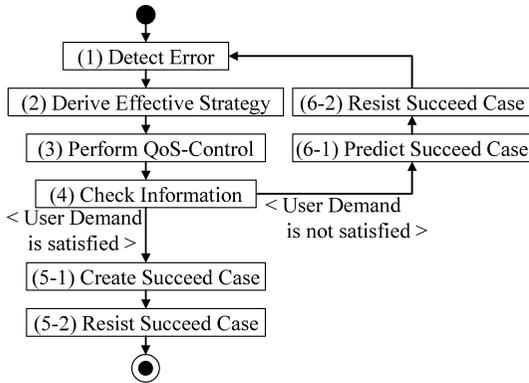


図 3 E-FVCS における QoS 調整の動作手順
Fig. 3 Flowchart of QoS-control in E-FVCS.

- (1) [問題検出] 計算機資源の利用状態が利用者要求を満たしていないことを E-FVCS が発見する。
- (2) [QoS 調整戦略決定] QoS 調整戦略決定知識を用いて、効果的と思われる QoS 調整戦略を推論する。
- (3) [QoS 調整] 手順 (2) で推論された QoS 調整戦略に従って QoS を調整する。
- (4) [QoS 調整終了] QoS 調整により利用者要求が満たされているか確認する。
- (5) [QoS 調整成功] 利用者要求が満たされていれば、問題検出 (手順 (1)) 時の情報と QoS 調整 (手順 (3)) で実行した QoS 調整戦略を組にして事例とする。
- (6) [QoS 調整失敗] 利用者要求が満たされていなければ、QoS 調整終了 (手順 (4)) 時の情報を用いて、問題検出 (手順 (1)) 時に最も効果的だった QoS 調整戦略を導出する。そして、問題検出 (手順 (1)) 時の情報と導出した効果的 QoS 調整戦略を組にして事例とする。

以上の手順により E-FVCS は QoS 調整を行い、また、QoS 調整戦略決定知識を獲得するための事例を収集する。

2.4 QoS 調整戦略の追加による計算処理量の増加

E-FVCS は QoS 調整の成功時、失敗時ともに事例を生成する。QoS 調整成功時は動作履歴を事例とするだけでよいが、失敗時は動作履歴から効果的 QoS 調整戦略を導出する必要があり、その計算処理量は E-FVCS に登録されている QoS 調整戦略の数に比例する。

図 4 に、QoS 調整失敗時において、E-FVCS が動作履歴から効果的 QoS 調整戦略を導出する手順を示す。この導出手順は、2.3 節で述べた QoS 調整動作

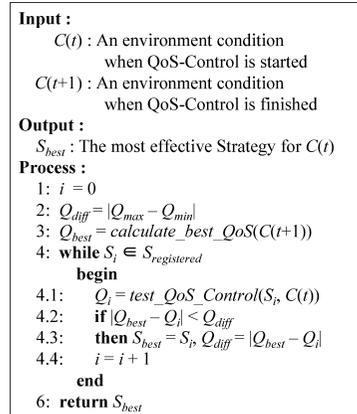


図 4 従来の E-FVCS における効果的 QoS 調整導出手順
Fig. 4 Algorithm for deriving an effective QoS-control-strategy in traditional E-FVCS.

の QoS 調整失敗 (手順 (6)) 時において実行されるが、その直前に実行された問題検出 (手順 (1)) 時の時刻を t 、また、QoS 調整終了 (手順 (4)) 時の時刻を $t+1$ とする。つまり、図 4 に示す導出手順は時刻 $t+1$ 以降に実行される。ここで、 Q_{max} は E-FVCS が提供可能な QoS の最高値、 Q_{min} は E-FVCS が提供可能な QoS の最低値であり、 $S_{registered}$ は E-FVCS に登録されている QoS 調整戦略の集合を表す。また、 $calculate_best_QoS(C(t+1))$ は、E-FVCS の動作環境の状況 $C(t+1)$ において最も理想的な QoS を導出する関数であり、 $test_QoS_Control(S_i, C(t))$ は、QoS 調整戦略 S_i と E-FVCS の動作環境の状況 $C(t)$ を用いて QoS 調整のシミュレート (QoS 調整テスト) を行う関数である。

E-FVCS の動作環境の状況 $C(t)$ は次の情報で構成される。

- (1) 動作環境の CPU 利用率や利用可能なネットワーク帯域などの計算機資源状態。
- (2) E-FVCS が提供している QoS、E-FVCS に対する利用者要求、および、E-FVCS による計算機資源の使用状況。
- (3) E-FVCS と計算機資源を共有しているアプリケーションの表示画面や利用者からの入力情報などの動作状態、および、このアプリケーションによる計算機資源の使用状況。

これらの情報は E-FVCS が動作中に履歴として収集する。

E-FVCS では、E-FVCS が提供可能な QoS の中で、提供サービスに対する利用者満足度が最も高くなる QoS を理想的な QoS (Q_{best}) とする。そのため、 Q_{best} を導出する関数 $calculate_best_QoS(C(t+1))$

では次の処理を行う。

- (1) $C(t+1)$ の状況で提供可能な QoS を導出する。
 - (2) 提供可能な QoS の中から、利用者の満足度が最も高い QoS を導出し、その QoS を Q_{best} とする。
- 処理 (1) の導出は QoS と必要計算機資源の関係知識⁸⁾ を用いて実現し、処理 (2) の導出は QoS と利用者満足度¹⁰⁾ の関係知識を用いて実現する。

QoS 調整テスト $test_QoS_Control(S_i, C(t))$ で行う処理を以下に示す。

- (1) 戦略 S_i に従って $C(t+1)$ を推論する。
- (2) 推論された $C(t+1)$ において提供可能な QoS を導出する。
- (3) 導出された QoS の中で利用者満足度の最も高い QoS を導出し、その QoS を返す。

処理 (2), (3) はすべての QoS 調整戦略に対しての共通処理であるが、処理 (1) はそれぞれの戦略によって異なる推論を行う。また、処理 (2), (3) の導出では、 $calculate_best_QoS(C(t+1))$ と同様に、QoS と必要計算機資源の関係知識、および、QoS と利用者満足度¹⁰⁾ の関係知識を用いて導出する。QoS 調整テストで推論すべき QoS は、E-FVCS の動作環境の状況 $C(t)$ に適応した QoS ではなく、次に観測される動作環境の状況 $C(t+1)$ に適応した QoS でなくてはならない。これは、 $test_QoS_Control(S_i, C(t))$ で推論された QoS が反映される動作環境の状態は $C(t+1)$ であるためである。

従来の E-FVCS では効果的 QoS 調整戦略を導出するために、E-FVCS に登録されているすべての QoS 調整戦略に対して QoS 調整テストを実行している。そのため、E-FVCS に登録される QoS 調整戦略が増加すると、効果的 QoS 調整戦略を導出するための計算処理量も増加する。より多くの環境に適応するためには、それらの環境に対応した QoS 調整戦略を追加登録する必要があるが、多くの QoS 調整戦略を E-FVCS に登録すると、効果的 QoS 調整戦略の導出に必要な計算機資源 (CPU 利用率など) が増加し、E-FVCS のサービスを提供するための計算機資源を圧迫し、E-FVCS が提供する音声や動画などの QoS が低下するという問題がある。

3. リスク値を用いた効率の良い効果的 QoS 調整戦略導出手法の提案

3.1 効果的 QoS 調整戦略導出手法の概要

E-FVCS における効果的 QoS 調整戦略導出の計算処理量を減らすためには、E-FVCS に登録されているすべての QoS 調整戦略に対して試験的に行う QoS 調

整のシミュレート (QoS 調整テスト) の実行回数を減らす必要がある。そこで本論文では、ある程度利用者要求を満足させる QoS を推論することができた QoS 調整戦略を発見した時点で QoS 調整テストを打ち切ることにより、効果的 QoS 調整戦略の計算処理量を軽減する手法を提案する。

効果的 QoS 調整戦略を導出するために行う QoS 調整テストの実行回数を減らすためには、効果的である可能性の高い QoS 調整戦略から QoS 調整テストを実行する必要がある。そのためには、過去における QoS 調整戦略の実行履歴を用いて、それぞれの QoS 調整戦略の効果的である可能性の高さを評価する必要がある。

この評価を行う手法として、実行頻度の高い QoS 調整戦略を効果的である可能性の高い戦略とする手法がある。しかし、この手法を用いて実行頻度の高い QoS 調整戦略から QoS 調整テストを実行した場合、実行頻度の高い戦略が数多く成功事例として登録されるため、この戦略の実行頻度がますます高くなるという現象が発生する。すなわち、効果的 QoS 調整戦略とされる戦略に偏りが発生し、獲得できる QoS 調整戦略決定知識が不正確になる問題がある。

この問題に対して本論文で提案する手法では、過去の QoS 調整における QoS 調整戦略の実行履歴、および、そのときの QoS 調整結果からリスク値を用いて QoS 調整戦略を評価する。リスク値の低い QoS 調整戦略を効果的である可能性の高い戦略として、リスク値の低い戦略から QoS 調整テストを実行する。提案手法を用いることにより、従来よりも少ない計算処理量で、従来と同程度に正確な QoS 調整戦略決定知識の獲得を実現する。

3.2 リスク値の計算方法

リスク値の求め方を以下に述べる。QoS 調整戦略 S_i の時刻 t におけるリスク値 $R(S_i, t)$ は

$$R(S_i, t) = \frac{\sum_{k=t-t_0}^t P(S_i, k)M(S_i, k)}{\sum_{k=t-t_0}^t P(S_i, k)} \quad (1)$$

として求められる。ここで、 $P(S_i, k)$ は時刻 k において QoS 調整戦略 S_i が実行される確率で、 $M(S_i, k)$ は時刻 k における QoS 調整戦略 S_i の失敗度である。また、 t_0 はリスク値のための観測期間を表している。 t はリスク値を算出する時刻であり、 $P(S_i, k)$ 、および、 $M(S_i, k)$ は E-FVCS の動作履歴より獲得する。 $P(S_i, k)$ は時刻 k における QoS 調整戦略決定知識により決定される。もし同時に実行される QoS 調整戦略が 1 個に限られている場合、 $P(S_i, k)$ は 0 か 1 の離散値をとり、複数の戦略が同時に実行される場合は

0 から 1 の連続値をとる．また， $P(S_i, k)$ の時間平均は QoS 調整戦略 S_i の実行頻度であり，QoS 調整戦略決定知識を変更すると，この実行頻度も変化する．

失敗度は，その QoS 調整戦略に従って QoS 調整を行ったときに，理想的な QoS からどれだけ離れているかを示しており，時刻 t に戦略 S_i を用いて行った QoS 調整結果の失敗度 $M(S_i, t)$ は，

$$M(S_i, t) = (Q(S_i, t) - Q_{best}(t))^2 \quad (2)$$

で求められる．ここで， $Q(S_i, t)$ は時刻 t に戦略 S_i に従って推論された QoS であり，E-FVCS の動作履歴から獲得する．また， $Q_{best}(t)$ は時刻 t において最も利用者を満足させる QoS であり，これは 2.4 節で述べた理想的な QoS (Q_{best}) と同じである．時刻 t に実行されなかった QoS 調整戦略の QoS 調整結果 $Q(S_i, t)$ は動作履歴に存在しないが，この戦略の時刻 t の実行確率 $P(S_i, t)$ が 0 となるため，リスク値の算出にこの結果は関与しない．すなわち， $Q(S_i, t)$ の獲得は E-FVCS の動作履歴からのみで十分である．

式 (1) で表されるリスク値 $R(S_i, t)$ は，観測期間に t_0 の間に発生した戦略 S_i による QoS 調整失敗度の期待値を示しており，リスク値が小さいほど，その戦略は的確な QoS 調整を行ってきたことを示している．

E-FVCS による QoS 調整の的確さを向上させるには，リスク値の小さい QoS 調整戦略の実行頻度が増加するように QoS 調整戦略決定知識を変更すればよいが，それぞれの QoS 調整戦略は万能ではなく，適応可能な状況と適応不可能な状況とがあることを考えると，実行頻度を過剰に増加させると QoS 調整戦略 S_i が適応不可能な状況においてもその戦略を採用してしまい，結果として $R(S_i, t)$ も増加することになる．リスク値の低い QoS 調整戦略の実行頻度が増加し，リスク値の高い QoS 調整戦略の実行頻度が減少し，

$$R(S_0, t) = R(S_1, t) = \dots = R(S_N, t) \quad (3)$$

に近づけば，すべての QoS 調整戦略がそれぞれに適した状況で実行されており，QoS 調整戦略決定知識は正確であると考えられる．

3.3 リスク値を用いた効果的 QoS 調整戦略の導出

図 5 にリスク値を用いて効果的 QoS 調整戦略を導出する手順を示す．従来の E-FVCS における効果的 QoS 調整戦略を導出する手順 (図 4) に対して，QoS 調整戦略をリスク値でソートする処理 (Process: 3 行目) と，QoS 調整テストの結果が利用者要求の許容範囲なら処理を終了する条件 (Process: 5 行目) を追加した． $sort_by_risk_value(S_{registered})$ では，E-FVCS に登録されている QoS 調整戦略 $S_{registered}$ をリスク値の昇順でソートする関数である．また， $Q_{tolerance}$ は提

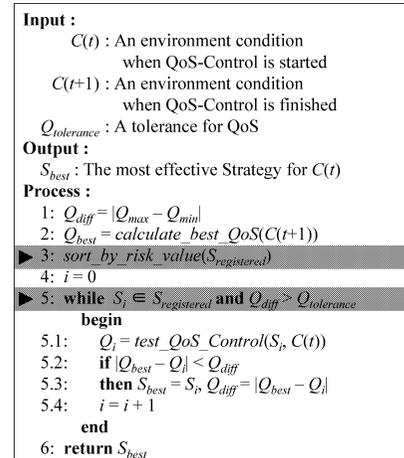


図 5 リスク値を用いた効果的 QoS 調整戦略導出手順

Fig. 5 Algorithm for deriving an effective QoS-control-strategy with risk-value.

供 QoS の許容誤差であり，E-FVCS の提供 QoS と理想的な QoS との差が $Q_{tolerance}$ 以内なら QoS 調整の必要がないと判断する．

提案手法における効果的 QoS 調整戦略の導出では，リスク値の低い QoS 調整戦略から QoS 調整テストを行い，利用者要求の許容範囲内の QoS 調整戦略を発見したところで処理を終了する．リスク値の低い QoS 調整戦略とは，E-FVCS の適応対象である計算機資源状態の変化に対して，的確な QoS 調整ができていた戦略である．リスク値の低い QoS 調整戦略から QoS 調整テストをして早期に効果的 QoS 調整戦略を発見し，QoS 調整テストの回数を減らすことにより，効果的 QoS 調整戦略の導出の計算処理量を減らしている．また，リスク値の低い QoS 調整戦略が数多く成功事例に登録されることにより，リスク値の低い QoS 調整戦略の実行頻度は増加し，リスク値の高い QoS 調整戦略の実行頻度は減少する．すなわち，これは式 (3) に近づくことを意味し，QoS 調整戦略決定知識は正確になるといえる．

我々が行ってきた研究³⁾ から，計算機資源状態が変化した場合，その変化に適応するために採用される QoS 調整戦略の割合は，計算機資源状態の変化に対して最も影響を与えたアプリケーションによって異なることが分かっている．そこで提案手法では，アプリケーションごとに QoS 調整戦略のリスク値を計算する．すなわち，計算機資源状態の変化に対して最も影響を与えたアプリケーションが異なると，QoS 調整戦略のリスク値も異なる．リスク値をアプリケーションごとに管理することにより，正確なリスク値を算出

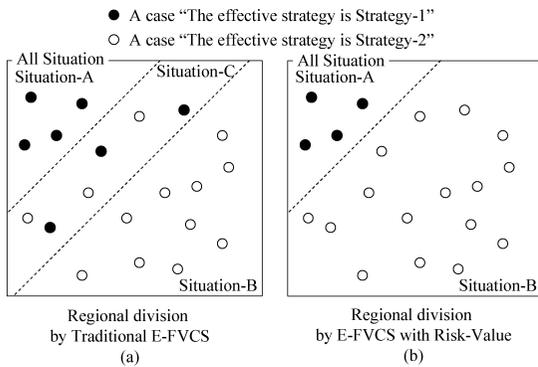


図 6 リスク値を用いることによる QoS 調整戦略決定知識の変化
Fig. 6 Change of knowledge to derive an effective QoS-control-strategy by using risk-value.

し、効率の良い効果的 QoS 調整戦略導出を実現する。

3.4 リスク値による QoS 調整戦略決定知識の簡潔化

図 6 に従来の E-FVCS と提案手法を適用した E-FVCS における、QoS 調整戦略決定知識の変化を示している。図 6 は Strategy-2 のリスク値が、Strategy-1 のリスク値より小さい場合を示している。E-FVCS では、蓄積された事例を分類することによりルールを生成し、それを QoS 調整戦略知識とする。たとえば、図 6 (a) では、

- Situation-A ならば、Strategy-1
- Situation-B ならば、Strategy-2
- Situation-C ならば、Strategy-1 もしくは Strategy-2

という QoS 調整戦略知識が獲得される。

Strategy-1 および Strategy-2 がともに適応できる状況が存在した場合、従来の E-FVCS では両方の QoS 調整戦略を同等に扱うため、2 つの QoS 調整戦略が効果的であると判断される状況 (図 6 (a): Situation-C) が存在した。しかし、提案手法では効果的 QoS 調整戦略として、リスク値の小さい QoS 調整戦略が数多く選択されるため、図 6 (a) の Situation-C においては Strategy-2 が効果的 QoS 調整戦略として導出される。以上の理由より、従来の E-FVCS では図 6 (a) となっていた QoS 調整戦略決定知識は、提案手法を適用した E-FVCS では図 6 (b) となる。

リスク値の大きい QoS 調整戦略とは、その戦略の適応範囲が狭く、また、適応できない状況では QoS 調整が機能しない特殊な戦略である。一方、リスク値の小さい QoS 調整戦略とは、その戦略の適応範囲が広く、また、適応範囲外の状況においても、ある程度の QoS 調整が期待できる汎用的な戦略である。提案

手法では、2 つ以上の QoS 調整戦略が適応できる状況に対しては、より汎用的な QoS 調整戦略を採用する。そのため、QoS 調整に失敗したときの失敗の大きさは従来の E-FVCS よりも小さくなる。また、QoS 調整戦略決定知識である QoS 調整戦略決定のためのルールは、従来の E-FVCS より簡潔となる。

4. 実験による評価

4.1 実験目的

本論文で提案した、リスク値を用いて効果的 QoS 調整戦略の導出を行う E-FVCS のプロトタイプシステムを用いて実験を行い、その結果を評価することにより本論文で提案した手法の有効性を検証する。本論文では提案手法を適用した E-FVCS を Proposed E-FVCS と呼ぶことにする。また、評価における比較対象として従来の E-FVCS³⁾ を用い、本論文ではこの E-FVCS を Traditional E-FVCS を呼ぶことにする。

4.2 プロトタイプ実装

提案手法の実現性を検証するために、プロトタイプシステムとして E-FVCS (Proposed E-FVCS, Traditional E-FVCS) を Java, および, C++ を用いて実装した。E-FVCS では、監視・適応する計算機資源は CPU 資源を対象とし、情報収集においては、アプリケーションごとの CPU 資源利用状態を収集対象とした。また、通信するメディアは動画メディアとし、調整する QoS パラメータは動画像のサイズとフレームレートとした。

QoS 調整戦略決定知識の獲得に関しては、事例からルールを生成することができる C4.5¹¹⁾ を用い、CPU 資源に余裕があるときにルール生成を行うように実装した。ルールに変換することにより QoS 決定時の推論コストを減らし、資源に余裕があるときにルール生成を行うことにより E-FVCS が提供するサービスへの影響を低減できる。

プロトタイプシステムには、最新のデータをもとに決定する戦略 (Agent1a, Agent1b), 過去数秒間の平均データを元に決定する戦略 (Agent2a, Agent2b, Agent2c), QoS を変化させない戦略 (Agent3a) や少しだけ変化させる戦略 (Agent3b), QoS を極端に変化させる戦略 (Agent3c) などを登録した。図 7 にプロトタイプシステムに登録した 8 個の QoS 調整戦略を示す。これらは、Web Browser や Compiler などの典型的なアプリケーションが CPU 資源を使用する状況を想定した戦略である。特定のアプリケーションへの対応や、CPU 資源以外の計算機資源に適応するためには、さらなる戦略の登録が必要である。実用的

Agent name	QoS-Control-Strategy characteristic
Agent1a	$Q(t+1) = k(r(t))$
Agent1b	$Q(t+1) = k\left(\frac{1}{2}r(t) + \frac{1}{2}r(t-1)\right)$
Agent2a	$Q(t+1) = k\left(\frac{1}{3}\sum_{i=0}^2 r(t-i)\right)$
Agent2b	$Q(t+1) = k\left(\frac{1}{7}\sum_{i=0}^6 r(t-i)\right)$
Agent2c	$Q(t+1) = k\left(\frac{1}{15}\sum_{i=0}^{14} r(t-i)\right)$
Agent3a	$Q(t+1) = Q(t)$
Agent3b	$Q(t+1) = \frac{1}{2}Q(t) + \frac{1}{2}k(r(t))$
Agent3c	if $r(t)$ is too high: $Q(t+1) = \min$

$q(t)$: QoS Parameter Value
 $r(t)$: CPU Usage of application without E-FVCS
 $k(r)$: CPU Usage to QoS Conversion

図 7 プロトタイプシステムに登録した QoS 調整戦略

Fig. 7 QoS-control-strategies registered for prototype system.

なシステムでは数十個の QoS 調整戦略を登録する必要があると思われる。

4.3 実験環境

実験では 2 つの計算機端末 Computer-A, および, Computer-B が 100 Mbps のネットワークで接続された環境で, それぞれの端末において E-FVCS を動作させて行った. Computer-A は CPU として Pentium2.4, メモリとして DDRRAM-512 MB を搭載し, Computer-B は CPU として Pentium2.8C, メモリとして DDRRAM-1 GB を搭載している. それぞれの端末には USB カメラを接続し, そのカメラから取得された動画をお互いに送受信する.

上述の実験環境において, Computer-A の CPU に負荷を与え, Computer-A の CPU 利用率と E-FVCS の QoS 調整の動作を観測する. Computer-A の CPU に与える負荷は, 15 個のアプリケーションを実際に使用したときの CPU 負荷を再現したものである.

本実験においての利用者要求は “計算機端末全体での CPU 利用率が 60%以上 80%以下” とした. よって, QoS 調整が必要である状況は

- (1) CPU 利用率が 80%より高く QoS を下げる必要がある場合,
- (2) CPU 利用率が 60%より低く QoS を上げる必要がある場合,

の 2 種類となる. また, リスク値計算に用いる失敗度は, QoS 提供時の CPU 利用率と理想的な CPU 利用率の差から求めることとした. すなわち, 失敗度 $M(S, t)$ は

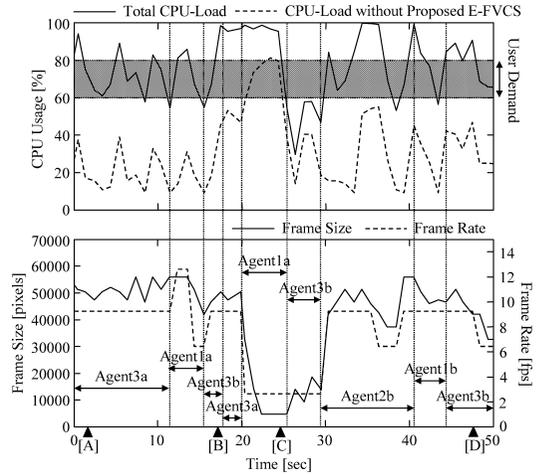


図 8 Proposed E-FVCS による QoS 調整

Fig. 8 QoS-control by Proposed E-FVCS.

$$M(S, t) = \begin{cases} (CPU\text{-Load} - 80\%)^2 & (\text{if } CPU\text{-Load} > 80\%) \\ (60\% - CPU\text{-Load})^2 & (\text{if } CPU\text{-Load} < 60\%) \end{cases} \quad (4)$$

となる.

4.4 実験結果

4.4.1 Proposed E-FVCS による QoS 調整動作

図 8 に Proposed E-FVCS の QoS 調整動作の結果を示す. 図 8 には, CPU 利用率の変化と, それに適応して Proposed E-FVCS が調整した QoS の変化, およびそのとき Proposed E-FVCS が用いた QoS 調整戦略について示している. E-FVCS 以外に CPU 資源を利用しているアプリケーションとして, 時刻 [A] から [B] の間では CPU を断続的に利用するアプリケーション (Web Browser) が動作し, 時刻 [B] から [C] の間は CPU を継続的に占有するアプリケーション (Installer) が動作し, 時刻 [C] から [D] の間では CPU を断続的に利用するアプリケーション (Web Browser) が動作している.

Proposed E-FVCS は QoS 調整が必要と判断すると, その場面に効果的である QoS 調整戦略を選択し, その戦略に従って QoS 調整を行う. 図 8 において, CPU を断続的に利用するアプリケーションが動作している場面 (時刻 [A]-[B], [C]-[D]) では, CPU 資源の一時的な枯渇が発生するが直後に回復することが望めるため, QoS を変化させない戦略 (Agent3a) や QoS を慎重に変化させる戦略 (Agent2b, Agent3b)

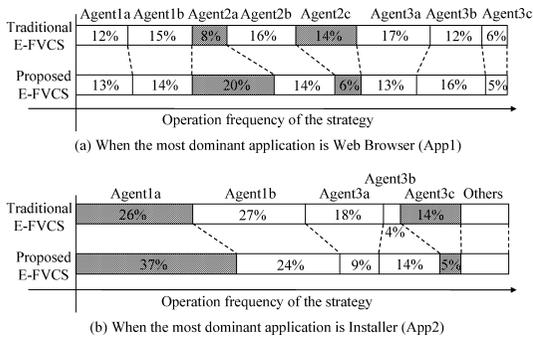


図 9 QoS 調整戦略の実行頻度

Fig. 9 Frequency of which each QoS-control-strategies are performed.

	Risk-Value	
	The most dominant application is Web Browser (App1)	The most dominant application is Installer (App2)
Agent1a	0.009	0.010
Agent1b	0.007	0.013
Agent2a	0.006	0.030
Agent2b	0.010	0.047
Agent2c	0.014	0.110
Agent3a	0.008	0.034
Agent3b	0.005	0.013
Agent3c	0.067	0.029

図 10 QoS 調整戦略のリスク値

Fig. 10 Risk-value of QoS-control-strategies.

を用いている。また、CPU を継続的に占有するアプリケーションが動作している場面（時刻 [B]–[C]）では、しばらく CPU 資源の枯渇状態が続くことが予想されるため、CPU 資源の不足に対応して QoS を低下させる戦略（Agent1a）を用いている。すなわち、Proposed E-FVCS は過去に蓄積した事例から QoS 調整戦略決定知識を獲得し、QoS 調整の場面对して効果的な QoS 調整戦略を選択することにより、的確な QoS 調整を実現している。

4.4.2 QoS 調整戦略の実行頻度の比較

図 9 に、Traditional E-FVCS、および Proposed E-FVCS におけるそれぞれの QoS 調整戦略の実行頻度を示す。図 9 (a) には CPU 資源の変化に対して最も影響を与えたアプリケーション（CPU 資源に対して支配的なアプリケーション）が Web Browser (App1) であった場合を示し、図 9 (b) には Installer (App2) であった場合を示している。また、図 10 には、Proposed E-FVCS が QoS 調整戦略の動作を観測することにより得られたリスク値を示している。いずれの結果も、Traditional E-FVCS、および Proposed E-FVCS を

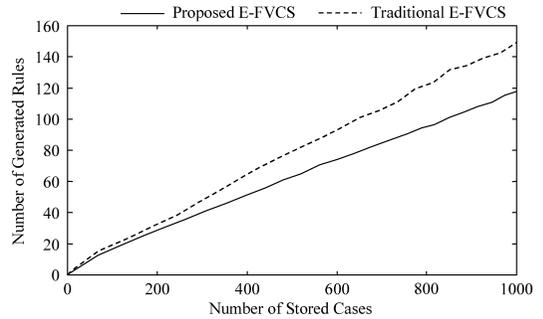


図 11 蓄積された事例数と生成されたルール数の関係

Fig. 11 A relationship between a number of stored cases and a number of generated rules.

24 時間以上動作させたときの結果である。

CPU 資源に対して支配的なアプリケーションが App1 であった状況（図 9 (a)）での、Proposed E-FVCS における Agent2a の実行頻度は、Traditional E-FVCS における実行頻度に対して大きい。また、Proposed E-FVCS における Agent2c の実行頻度は、Traditional E-FVCS における実行頻度に対して小さい。これは、Proposed E-FVCS における QoS 調整戦略の実行頻度が、図 10 に示されるリスク値の影響を受けているためである。CPU 資源に対して支配的なアプリケーションが App1 の場合、Agent2a のリスク値は小さいため、Agent2a は優先的に QoS 調整戦略テストの対象となる。そのため、この戦略が QoS 調整戦略決定知識獲得のための成功事例として登録される可能性が高くなり、実行頻度は増加する。また、Agent2c のリスク値は大きい値になっているため、この戦略が成功事例として登録される可能性は低くなり、実行頻度は低下する。

同様の理由で、CPU 資源に対して支配的なアプリケーションが App2 であった状況（図 9 (b)）においても、Proposed E-FVCS における Agent1a の実行頻度は Traditional E-FVCS のそれよりも大きくなり、Agent3c の実行頻度が小さくなっている。

4.4.3 QoS 調整戦略決定知識の簡潔さの比較

図 11 に Proposed E-FVCS と Traditional E-FVCS に蓄積された事例数と生成されたルール数の関係を示す。Proposed E-FVCS におけるルール生成数は、Traditional E-FVCS におけるルール生成数に比べて少ない。3.4 節で述べたように、2 つ以上の QoS 調整戦略が対応している場面において、従来の Traditional E-FVCS では、どの QoS 調整戦略を効果的な QoS 調整戦略とするか不定であったが、Proposed E-FVCS ではリスク値の最も小さい QoS 調整戦略を効果的な QoS 調整戦略とする。そのため Proposed E-FVCS

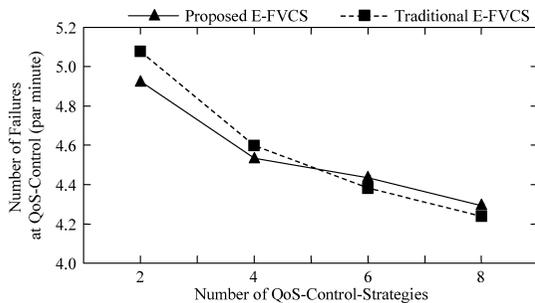


図 12 QoS 調整戦略数と QoS 調整的的確さの関係

Fig. 12 A relationship between accuracy of QoS-control and a number of QoS-control-strategies.

の QoS 調整戦略決定知識は, Traditional E-FVCS よりも簡潔なものとなっている。

4.5 評価

4.5.1 評価目的

提案手法が, 従来よりも少ない計算処理料で従来と同等に的確な QoS 調整を実現していることを検証するために, 以下の項目について評価を行った。

- (1) QoS 調整戦略の登録数と QoS 調整的的確さの関係
- (2) QoS 調整戦略の登録数と効果的 QoS 調整戦略導出の計算処理量の関係
- (3) QoS 調整戦略決定知識の獲得に要する時間
- (4) QoS 調整的的確さ

まず (1) により, QoS 調整的的確さを向上させるためには QoS 調整戦略を追加登録することが重要であることを示す。また (2) より, 提案手法が従来手法に比べて計算処理量が少ないことを検証する。さらに (3), (4) より, QoS 調整的的確さは従来と同等であることを示す。

本論文では, QoS 調整的的確さを 1 分間あたりの QoS 調整失敗数で評価する。QoS 調整失敗とは, QoS 調整を行ったが, その結果が利用者要求を満足しない場合を指す。的確な QoS 調整ができていれば, QoS 調整失敗数は減少する。

4.5.2 QoS 調整戦略の登録数と QoS 調整的的確さの関係

図 12 に, E-FVCS に登録されている QoS 調整戦略の数と QoS 調整的的確さの関係を示す。Proposed E-FVCS, Traditional E-FVCS とともに, 登録される QoS 調整戦略が増加すると, QoS 調整はよりの確になる。これは, QoS 調整戦略が追加登録されると, その戦略が対応している計算機資源状態の変化に適應できるようになるためである。

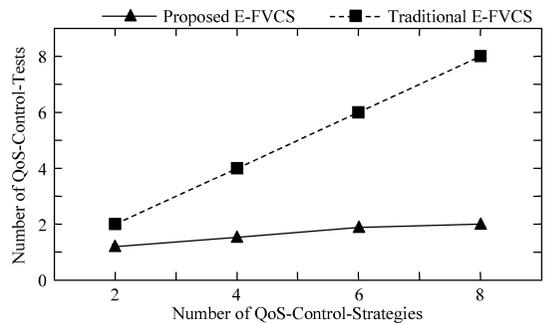


図 13 効果的 QoS 調整戦略導出のための QoS 調整テストの実行回数

Fig. 13 Number of QoS-control-tests to derive an effective QoS-control-strategy.

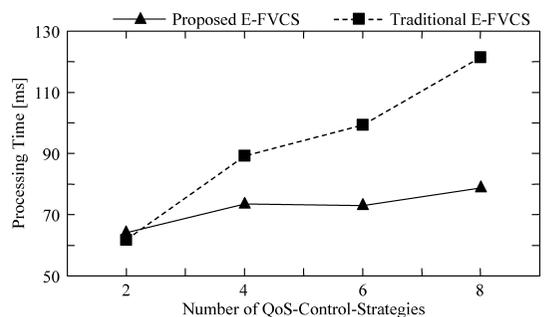


図 14 効果的 QoS 調整戦略導出の実行時間

Fig. 14 Processing time to derive an effective QoS-control-strategy.

4.5.3 QoS 調整戦略の登録数と効果的 QoS 調整戦略の導出コストの関係

図 13 に, 登録されている QoS 調整戦略の数と効果的 QoS 調整戦略を導出するために実行される QoS 調整テストの回数の関係を示す。また, 図 14 に, 登録されている QoS 調整戦略の数と効果的 QoS 調整戦略導出のための計算時間の関係を示す。図 13, および, 図 14 は, Proposed E-FVCS および Traditional E-FVCS について, それぞれを 180 分間動作させ, その間に行われた効果的 QoS 調整戦略の導出処理について測定したものである。なお, Traditional E-FVCS の測定結果より, 1 回の QoS 調整テストに要する計算時間は 10 ms 程度であると考えられる。

QoS 調整戦略の登録数の増加に対して, Proposed E-FVCS における QoS 調整テスト回数の増加は Traditional E-FVCS における増加量の 10%程度であり, Proposed E-FVCS における計算処理時間の増加は Traditional E-FVCS における増加量の 20%程度であった。すなわち, Proposed E-FVCS に登録可能な QoS 調整戦略の数は Traditional E-FVCS の 5 倍程度であると考えられる。

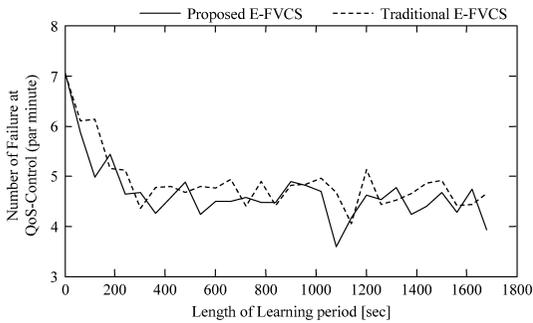


図 15 QoS 調整戦略決定知識の獲得時間の長さ と QoS 調整の的確さの関係

Fig. 15 A relationship between accuracy of QoS-control and length of learning period.

	Number of Failure at QoS-Control (par 1minute)	Average of QoS adjustment error (Average of $\sqrt{M(S,t)}$)
Traditional E-FVCS	4.68	0.044
Proposed E-FVCS	4.67	0.039

図 16 Proposed E-FVCS と Traditional E-FVCS における QoS 調整の的確さ

Fig. 16 Accuracy of QoS-control in Proposed E-FVCS and Traditional E-FVCS.

4.5.4 QoS 調整戦略決定知識の獲得に要する時間

図 15 に QoS 調整戦略決定知識の獲得時間の長さ と QoS 調整の的確さの関係を示す。図 15 は、それぞれ 50 回測定したときの平均値を示している。

Proposed E-FVCS, および Traditional E-FVCS はともに、知識獲得時間が長くなるに従い、よりの確な QoS 調整を行っているが、その的確さの変化は、Proposed E-FVCS, Traditional E-FVCS とともに同様な変化となった。このことより、QoS 調整戦略決定知識の獲得に要する時間に対する、提案手法の影響はないといえる。

4.5.5 QoS 調整の的確さ

図 16 に Proposed E-FVCS と Traditional E-FVCS を 24 時間以上動作させたときの QoS 調整の的確さの比較を示す。ここでは QoS 調整の的確さについて、1 分間あたりの QoS 調整失敗数だけでなく、QoS 調整後に E-FVCS から提供された QoS と理想的な QoS との差の平均値も用いる。また、ここでは理想的な QoS との差を QoS 調整後の CPU 利用率と理想的な CPU 利用率の差とした。これは式 (4) で表される失敗度の平方根となる。QoS 調整失敗数について Proposed E-FVCS と Traditional E-FVCS では明確な差は確認できなかった。一方で、Proposed E-FVCS の提供する QoS と理想的な QoS との差の平均は、Traditional E-FVCS のそれよりも小さくなっ

た。これは図 9 で示したように、Proposed E-FVCS ではリスク値の大きい QoS 調整戦略の実行頻度が減少したため、それらの戦略による QoS 調整失敗数が減少したためであると考えられる。

4.5.6 議 論

4.5.3 項で、提案手法を適用した Proposed E-FVCS は、従来手法である Traditional E-FVCS より少ない計算処理量で効果的 QoS 調整戦略を導出でき、また、Proposed E-FVCS に QoS 調整戦略を追加登録したときの効果的 QoS 調整戦略導出のための計算処理量の増加は、Traditional E-FVCS の 20% 程度であることを示した。これより Proposed E-FVCS に新たな QoS 調整戦略を追加登録することは Traditional E-FVCS よりも容易であり、Proposed E-FVCS に登録可能な QoS 調整戦略の数は Traditional E-FVCS の約 5 倍であるといえる。E-FVCS に登録される QoS 調整戦略の数は、対応できる計算機資源状態の変化特性の数と同意である。よって、Proposed E-FVCS は Traditional E-FVCS と比較して、より多くの種類の計算機資源状態の変化特性に対応でき、従来に比べて多様な環境において的確な QoS 調整が可能である。

4.5.5 項では、Proposed E-FVCS の QoS 調整失敗数は Traditional E-FVCS と同等であり、Proposed E-FVCS の QoS 調整の的確さは Traditional E-FVCS と同等であることを示した。また、Proposed E-FVCS が提供する QoS と理想的な QoS との差の平均は Traditional E-FVCS のそれよりも小さいことを示した。このことから、Proposed E-FVCS での QoS 調整失敗は、Traditional E-FVCS でのそれに比べて軽度であり、つまり、QoS 調整失敗時に Proposed E-FVCS が利用者を与える不快感は、Traditional E-FVCS のそれよりも小さいといえる。

5. 検 討

5.1 提案手法の適用範囲

本論文で述べたプロトタイプシステムでは適応対象の計算機資源を CPU とした。CPU 以外に適応可能な計算機資源としては、ADSL などのアクセス回線の伝送帯域などの少数の利用者によって占有される計算機資源があげられる。適応対象の計算機資源として、これらの計算機資源が追加されたとしても、それぞれの計算機資源の変化に対して各個に対応すればよいため、提案手法は有効であると考えられる。ただし、CPU 以外の計算機資源の変化に適応するためには、その計算機資源の変化特性に対応した QoS 調整戦略を登録する必要があり、どのような戦略が必要になるかは今後の

課題である。また、E-FVCS はインターネットのバックボーン回線の伝送帯域などの多くの利用者が共有する計算機資源には適応できない。これは、E-FVCS における QoS 調整戦略決定知識の獲得では「計算機資源に対して同時に影響を与えるアプリケーションは少数である」という仮定を用いており、この仮定が成立しない計算機資源に対しては適応できないためである。これらの計算機資源に対する適応も今後の課題である。

5.2 マルチメディア通信システムへの応用

本論文ではビデオ会議システムを取り上げ、そのシステムにおける提案手法の有効性を検証した。ただし、ビデオ会議システムだけではなく、IP 電話や多人数参加型ゲームなどのマルチメディア通信システム (MCS) において的確な QoS 調整を行うためには、それぞれの動作環境で効果的な QoS 調整戦略を登録する必要がある。そのため、これらの MCS でも QoS 調整戦略の登録数増加にともなう計算処理量増加の問題は発生し、この問題解決のために提案手法は有効であると考えられる。

また既存研究として、複数のポリシーを切り替えることにより動作環境に適応した機能調整を実現するミドルウェアシステムがある¹²⁾。このシステムではポリシーの増加にともない、効果的なポリシーの導出に必要な計算処理量が増加するという問題がある。本論文で提案したリスク値を用いてそれぞれのポリシーを評価し、その評価に基づいてポリシーの検証順序を変更することにより、ポリシー導出の計算処理量の減少が期待できる。

6. む す び

本論文では、計算機資源状態の変化に適応して的確な QoS 調整を実現するマルチメディア通信システムである E-FVCS において、効果的 QoS 調整戦略の計算処理量を減らすことを目的とし、リスク値を用いて効果的 QoS 調整戦略を効率良く導出する手法を提案した。具体的には、効果的な QoS 調整戦略を導出するために行う QoS 調整テストの回数を減らすために、リスク値を用いてそれぞれの QoS 調整戦略を評価する手法、およびその評価に基づいた効果的 QoS 調整戦略を導出する手法について提案した。また、提案手法を適用した E-FVCS を用いて実験を行い、実験結果より提案手法が、従来より少ない計算処理量で、従来と同等に的確な QoS 調整を実現していることを確認した。提案手法を用いることにより、従来より多くの QoS 調整戦略を E-FVCS に登録することが可能となり、より多様な環境において的確な QoS 調整が可能となる。すなわち、従来より多くの環境において、

MCS が計算機資源を占有して他のアプリケーションが動作できない状況や、利用可能な計算機資源に対して MCS が提供する QoS が低すぎる状況を減らすことが可能となる。

今後の課題としては、QoS 調整の結果を評価することにより、計算機資源状態を把握する手法を考案することがあげられる。QoS 調整戦略を決定するためには、計算機資源状態を正確に把握することが重要であるが、計算機資源状態を把握するためには計算機資源の情報だけではなく、そのとき行われた QoS 調整の結果も有用であると思われる。本論文では、QoS 調整の結果を QoS 調整戦略に対する評価として活用した。しかし、QoS 調整の結果を計算機資源状態を把握するためにも活用することにより、より正確な QoS 調整戦略決定知識を獲得することも可能であると考えられ、この知識獲得を実現することが今後の課題である。

参 考 文 献

- 1) Suganuma, T., Lee, S., Kinoshita, T. and Shiratori, N.: An Agent Architecture for Strategy-centric QoS Control in Flexible Videoconference System, *New Generation Computing*, Vol.19, pp.173-191 (2001).
- 2) Takeda, A., Uchiya, T., Kitagata, G., Suganuma, T., Kinoshita, T. and Shiratori, N.: Effective QoS control mechanism for flexible videoconference system, *Proc. 1st International Forum on Information and Computer Technology (IFICT2003)*, pp.160-165 (2003).
- 3) 武田敦志, 打矢隆弘, 北形 元, 菅沼拓夫, 白鳥則郎: やわらかいビデオ会議システムにおける効果的な QoS 調整戦略を決定するための知識獲得手法, 合同エージェントワークショップ&シンポジウム 2004, pp.157-164 (2004).
- 4) 武田敦志, 打矢隆弘, 北形 元, 菅沼拓夫, 白鳥則郎: やわらかいビデオ会議システムにおける効果的な QoS 調整戦略を決定するための知識獲得手法, 電子情報通信学会論文誌, Vol.J88-D-1, No.9, pp.1378-1387 (2005).
- 5) Braden, B., Zhang, L., Estrin, D., Herzog, S. and Jamin, S.: Resource reservation protocol (RSVP) — version 1 functional specification, RFC2205 (1997).
- 6) 阿部 睦, 鳩野淳子, 渥美幸雄: 端末におけるサービス品質保証方式の提案と評価, 電子情報通信学会論文誌 B, Vol.J82-B, No.5, pp.711-721 (1999).
- 7) 加藤由花, 箱崎勝也: 分散マルチメディアシステムにおけるアプリケーション QoS 管理手法, 情報処理学会論文誌, Vol.42, No.10, pp.2494-2504 (2001).

- 8) 中岡 謙, 松田 潤: ファジークラシファイアシステムによる QoS マッピングルールの獲得, 電子情報通信学会論文誌 D-1, Vol.J85-D-1, No.1, pp.69-78 (2002).
- 9) Ruiz, P.M., Botia, J.A. and Gomez-Skarmeta, A.: Providing QoS Through Machine-Learning-Driven Adaptive Multimedia Applications, *IEEE Trans. Systems*, Vol.34, No.3, pp.1398-1411 (2004).
- 10) Hands, D.S.: A Basic Multimedia Quality Model, *IEEE Trans. Multimedia*, Vol.6, No.6, pp.806-816 (2004).
- 11) Quinlan, J.: *C4.5: Programs for Machine Learning*, Morgan Kaufmann (1993).
- 12) Capra, L., Emmerich, W. and Mascolo, C.: CARISMA: Context-Aware Reflective mIddleware System for Mobile Applications, *IEEE Trans. Softw. Eng.*, Vol.29, No.10, pp.929-944 (2003).

(平成 17 年 5 月 19 日受付)

(平成 17 年 11 月 1 日採録)



武田 敦志

1977 年生 . 2005 年東北大学大学院情報科学研究科博士後期課程修了 . 2005 年東北文化学園大学科学技術学部応用情報工学科助手 . 博士 (情報科学) . やわらかいネットワーク, エージェント指向コンピューティングに興味を持つ .



今井信太郎 (学生会員)

2002 年東北大学工学部情報工学科卒業 . 2004 年同大学大学院情報科学研究科博士課程前期修了 . 現在, 同大学院情報科学研究科博士課程後期在学中 . エージェント指向コンピューティングに関する研究に従事 .



北形 元 (正会員)

1972 年生 . 2002 年東北大学大学院情報科学研究科博士後期課程修了 . 現在, 東北大学電気通信研究所助手 . 博士 (情報科学) . エージェント指向コンピューティング, エージェント型ネットワークミドルウェアに興味を持つ . 電子情報通信学会会員 .



菅沼 拓夫 (正会員)

1966 年生 . 1997 年千葉工業大学大学院博士後期課程修了 . 1997 年東北大学電気通信研究所助手 . 2003 年同研究所助教授 . やわらかいネットワーク, エージェント指向コンピューティング, 共生コンピューティング等の研究開発に従事 . Th 8th JWCC Best Presentation Award 等受賞 . 博士 (工学) . IEEE, 電子情報通信学会各会員 .



木下 哲男 (正会員)

1979 年東北大学大学院修士課程修了 . 同年沖電気工業 (株) 入社 . 知識情報処理技術の研究開発に従事 . 1996 年東北大学電気通信研究所助教授, 2001 年同大学情報シナジーセンター教授 . 知識工学, エージェント工学, 知識応用システム等の研究開発に従事 . 本会 1989 年度研究賞, 1996 年度論文賞, 電子情報通信学会 2001 年度業績賞等受賞 . 工学博士 . 電子情報通信学会, 人工知能学会, 日本認知科学会, IEEE, ACM, AAAI 各会員 .



白鳥 則郎 (フェロー)

1946 年生 . 1977 年東北大学大学院博士課程修了 . 1984 年同大学助教授 (電気通信研究所) . 1990 年同大学教授 (工学部情報工学科) . 1993 年同大学教授 (電気通信研究所) . 情報通信システム, 人と IT 環境の共生の研究に従事 . 1993 年本会マルチメディアと分散処理研究会主査 . 本会 25 周年記念論文賞 . 本会副会長, 本会フェロー, IEEE フェロー, 電子情報通信学会フェロー .