Vol.2010-DPS-142 No.24
Vol.2010-CSEC-48 No.24
2010/3/4

システムコールのタイミング制御による コミュニケーション品質劣化防止方式

金友大　　中島一彰
NEC サービスプラットフォーム研究所
E-mail: d-kanetomo@ce.jp.nec.com

PC 上のソフトウェアで動作する電話であるソフトフォンでは、通話中に PC 上でほかのアプリケーションを利用すると、PC が過負荷になることで、音切れなどの音声品質の劣化が発生してコミュニケーションが中断する問題がある。音声品質の劣化を回避するため、OS が提供するプロセス優先度制御でソフトフォンのプロセスに高い優先度を設定した場合でも、ほかの競合するアプリケーションのシステムコール実行により、最高優先度の OS による処理が発生する際には、音質劣化が避けられないことが実験により判明している。本論文では、この問題を解決するため、競合アプリケーションのシステムコールのタイミングを制御することで、ソフトフォンの音声品質を維持するコミュニケーション品質劣化防止方式を提案する。評価実験により PESQ 値が平均 3.2 から 4.0 に改善したことで、提案方式が有効であることを示す。

Preventing deterioration of communication QoE by controlling interval of system calls

Dai Kanetomo and Kazuaki Nakajima

On communication with a softphone, we are often annoyed by deterioration of communication QoE because several other applications are simultaneously running on a PC during the communication. The deterioration of QoE is not prevented by priority setting of each process. This paper proposes a method for preventing deterioration of communication QoE by controlling interval of system call execution, and shows effectiveness of the method through experimentation.

1. はじめに

ネットワークに接続された PC の能力向上により、PC を電話や電子会議などのリアルタイムコミュニケーションの端末として利用することが一般的になってきている。特に企業でのコミュニケーションツールとして、ソフトフォンと呼ばれるソフトウェアが利用され、PC にヘッドセットを接続しての音声通話に加えて、Web カメラを接続したテレビ電話や、データ共有ツールと連動してのオンライン会議などが可能になっている。企業でのソフトフォン導入時には、新たなハードウェア導入コストの削減や PC 上のほかのアプリケーションとの連携のため、すでに業務で利用している PC にソフトフォンを追加で導入するケースが多い。このような導入形態では、ソフトフォンを利用したコミュニケーション中にも、様々なアプリケーションが同時に PC 上で実行されている。例えば音声通話をしながら関連情報を検索するための Web ブラウザや、過去のメールを参照するためのメールクライアントを同時に PC 上で利用することが考えられる。

また、企業内ではアンチウイルスソフトウェアなどセキュリティ関連のソフトウェアが PC 上に常駐し、定期的に監査処理をしている。ソフトフォン利用中にこのようなアプリケーションが同時に実行されると、ソフトフォンのデータ処理がほかの同時実行アプリケーションに妨げられて、コミュニケーションの品質劣化が発生する。例えば、コミュニケーションの基本である音声通話については、音切れが発生することにより相手の発言内容が聞き取れなくなり、コミュニケーションが成り立たなくなる。

本稿ではソフトフォンでのコミュニケーション中に PC 上でほかのアプリケーションを同時に利用した場合でも、コミュニケーションの品質劣化の発生を抑制するコミュニケーション品質劣化防止方式を提案する。

2. コミュニケーション品質劣化防止方式の要件と課題

2.1 コミュニケーション品質劣化防止方式の要件

コミュニケーション品質劣化防止方式は次の 2 つの要件を満たさなければならない。1 つはソフトフォンが必要とするタイミングでコミュニケーションに関するデータ処理を実行可能にすることである。ソフトフォンを含む複数のアプリケーションが同時に PC 上で実行される場合、OS が各アプリケーションの処理実行タイミングを制御するが、複数のアプリケーションの処理要求が重なった場合には、いずれかのアプリケーションの処理実行が遅れる。ソフトフォンの音声通話の処理は、ストリームデータの受信と再生を連続して行うので、処理の遅れに弱く、音切れなどの音質劣化が発生しやすい。通信データのバッファリングで処理の遅れをある程度カバーできるが、音声通話のようなリアルタイムコミュニケーションでは通話の遅延につながるため長時間のバッファリングは困難であるので、データ処理を連続的に実行しなければならない。

もう 1 つの要件はソフトフォンと同時に実行するアプリケーションを完全に停止させないことである。不要なアプリケーションは停止させることがソフトフォンのデータ処理を妨げない最も簡単な方法だが、同時に実行するアプリケーションは様々な面

Vol.2010-DPS-142 No.24
Vol.2010-CSEC-48 No.24
2010/3/4

でソフトフォンでのコミュニケーションをサポートしている。コミュニケーションに関連する情報を検索するための Web ブラウザやメールクライアントなど、ユーザが意図してコミュニケーションを効率化するために利用するアプリケーションだけでなく、セキュリティ関連のアプリケーションを含むバックグラウンドプロセスも、PC を正常な状態に保つために稼動していることから、コミュニケーションをサポートしていると考えられる。

2.2 従来技術の課題

アプリケーションが必要なタイミングで処理を実行可能にするための従来技術は「優先度設定」と「リソース予約」の 2 つに大別できる。優先度設定方式は、ユーザが予めアプリケーションのプロセスに優先度を設定すると、複数のプロセスを並行して実行される場合に、OS が優先度の高いプロセスの処理要求を優先して CPU で処理させる方式である。例えば Windows XP ではリアルタイム、高、通常以上、通常、通常以下、低の 6 種類の優先度が定義されており、優先度が「高」のプロセス A が処理を要求すると、優先度「通常」のプロセス B が処理中であっても、プロセス A の処理に切り替わる**[1]**。本方式を適用した場合、例えば、**表 1** のように設定することで、ある程度コミュニケーション品質の劣化を抑制できるが、音切れによるコミュニケーション品質劣化を完全に回避することはできない。なぜなら、アプリケーションがハードディスクやネットワークカードなどのデバイスへのアクセスなどで OS が提供するシステムコールを利用すると、システムコールによって実行される機能がカーネルモードで動作し、アプリケーションの優先度設定に関係なく常に最高優先度で実行されるからである。OS 機能が最優先で処理されることで、ソフトフォンの処理が遅延すると音切れが発生する可能性がある。特にシステムコールが連続的に実行されると、ソフトフォンの処理遅延時間が長くなるため、音切れの発生する可能性が増大する。

リソース予約方式には、例えば CPU Broker のように、各アプリケーションが必要とする CPU 時間を調停役のプロセスに要求すると、調停役のプロセスが重複する要求に対しては調整をした後に、OS の提供する API を使って CPU 処理時間を各アプリケーションに割り当てる方式がある**[2]**。リソース予約方式のもう一つの例としては Dynamic QoS Resource Manager (DQM) のように、調停役のプロセスが利用可能な CPU 時間内に各アプリケーションが必要とする CPU 時間を収めるために、各アプリケーションの API を使ってアプリケーションの処理速度を制御する方式がある**[3]**。これらの 2 つの方式は OS が提供する CPU 時間の設定 API や、各アプリケーションの処理速度を制御する API を前提としており、適用が困難である。なぜなら CPU 時間の設定 API は組み込み用 OS には存在するが PC 用の OS には存在せず、また PC 上でソフトフォンと同時に実行される可能性のあるアプリケーションの種類は様々で、特定の API の存在を前提とすることは困難なためである。

表 1 プロセス優先度設定の例

アプリケーション	ソフトフォン	アンチウイルス ソフトウェア	そのほかの アプリケーション
プロセス優先度	高	通常以上	通常

3. 提案するコミュニケーション品質劣化防止方式

3.1 前提とする実行環境

本稿で提案するコミュニケーション品質劣化防止方式の設計において、次の 2 つの前提を置いている。

1) ソフトフォンおよび提案する品質劣化防止方式の実行環境である PC の OS は Windows であること
2) ソフトフォンと同時に実行されるアプリケーションに特別な前提を必要としないこと

PC の OS として Windows のシェアは 88.4%であり、最も一般的な実行環境である**[4]**。また、ソフトフォンと同時に実行される可能性のあるアプリケーションは様々であり、特別な前提を置くことは困難である。そこで提案する品質劣化防止方式は OS の機能を利用して実現可能な方式として設計した。

3.2 品質劣化防止方式の概要

図 1 を参照して品質劣化防止方式の概要を説明する。**図 1** ではソフトフォンと同時に実行されるアプリケーション（同時実行アプリ）のシステムコールを OS に対する上から下への矢印で表現しており、矢印が OS に達するとシステムコールに対応した OS 機能が実行されることを意味する。2.2 節で従来技術である優先度制御の課題として述べたとおり、a のように連続的にシステムコールが実行されると、最高優先度の OS での処理が長い時間行われて、音切れなどの音質劣化が認識されるほどソフトフォンの音声処理遅延が大きくなる。これを防止するためにはシステムコール実行を止める必要はなく、実行タイミングだけを制御し、システムコールの実行と実行の間にソフトフォンのデータ処理が実行できるようにすればよい。

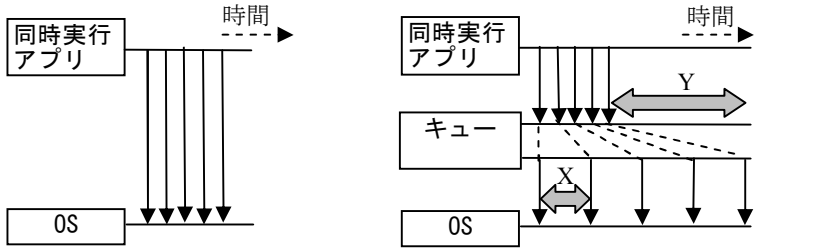
そこで提案方式としてはシステムコール実行頻度（一定時間内のシステムコール実行回数）に閾値を設け、システムコールの実行頻度が閾値以上になった場合には、b に示すように同時実行アプリによるシステムコールを OS で実行する前に一旦キューイングし、一定時間間隔（図中の X）で実行されるように実行タイミングを制御する。一方で連続して実行されるはずのシステムコールをキューイングして一定時間間隔で実行するように単純に制御した場合、キューの最後のシステムコールはアプリケーションでのシステムコール利用から OS での実行までに時間がかかる（図中の Y）。アプリケーションはデバイスのトラブルなどで処理が停止してしまうことを避けるために、

Vol.2010-DPS-142 No.24
Vol.2010-CSEC-48 No.24
2010/3/4

システムコールの実行をタイマで監視している場合があるため、処理実行までに時間がかかるとアプリケーションがシステムコール利用をタイムアウトさせ、本来の意図した処理が実行されない可能性がある。

この問題を防ぐためには、アプリケーションでのシステムコール利用から OS での実行までの時間待ち時間を一定時間以内にする必要がある。そこで実行待ちシステムコール数に応じてシステムコール実行間隔（図中の X）を動的に制御し、最も長い実行待ち時間（図中の Y）でも予め与えた実行待ち上限時間内にシステムコールが実行されるようにする。

アプリケーションが利用するシステムコールは種類が多いが、それぞれのシステムコールにより OS での実行負荷（時間）が異なるため、必ずしも全てのシステムコール実行を制御しなければならないわけではない。音質劣化を引き起こすシステムコールを特定し、それだけを制御対象とすることで、元のアプリケーションの実行への影響を最小限にすることができる。本方式の具体的な実現方法について次節以降でさらに詳しく述べる。



a. システムコール連続実行　　b. システムコール実行タイミング制御

図 1 システムコールの実行タイミング

3.3 機能構成と各機能の振る舞い

本稿で提案するコミュニケーション品質劣化防止方式は、**図 2** に示すような機能構成で実現することができる。カウンタ機能は同時実行アプリと 1:1 対応で存在し、対応する同時実行アプリケーションのシステムコール実行状況を監視し、実行回数をカウントする。カウントした結果はカウンタ集計機能によって定期的に集計される。これにより PC 全体でのシステムコール実行頻度を計測することが可能になる。

カウンタ集計機能が計測したシステムコール実行頻度に基づいて、制御内容決定機能がシステムコール実行タイミング制御の要・不要の判断と、制御時のシステムコール実行間隔時間を決定する。システムコール実行タイミング制御は予め与えられたシステムコール利用頻度の閾値を越えた場合に必要と判断され、制御内容決定機能から実行タイミング制御機能に対し、制御開始が指示される。

実行タイミング制御機能はカウンタ機能と同様に同時実行アプリと 1:1 で存在する。同時実行アプリがシステムコールを利用すると、実行タイミング制御機能はシステムコールが OS で実行される前にキューイングし、制御内容決定機能にシステムコール利用があったことを通知する。制御内容決定機能は一定時間間隔でシステムコールが実行されるように、同時実行アプリによる利用順に実行タイミング制御機能に対し、OS でのシステムコール実行を指示する。制御内容決定機能は制御開始時点では予め与えられた時間間隔で実行を指示するが、キューイングにより実行待ちのシステムコール数が増加するたびに、キューイングによる最大実行待ち時間を計算する。最大実行待ち時間が予め与えられた実行待ち上限時間を超える場合には、キューイングによる最大実行待ち時間が実行待ち上限時間に収まるようにシステムコールの実行間隔を調整する。

キューイングされた実行待ちシステムコールがなくなった場合には、制御内容決定機能はシステムコール実行タイミング制御が必要な状態は終了したと判断し、実行制御を停止し、システムコール実行を監視する状態に戻る。

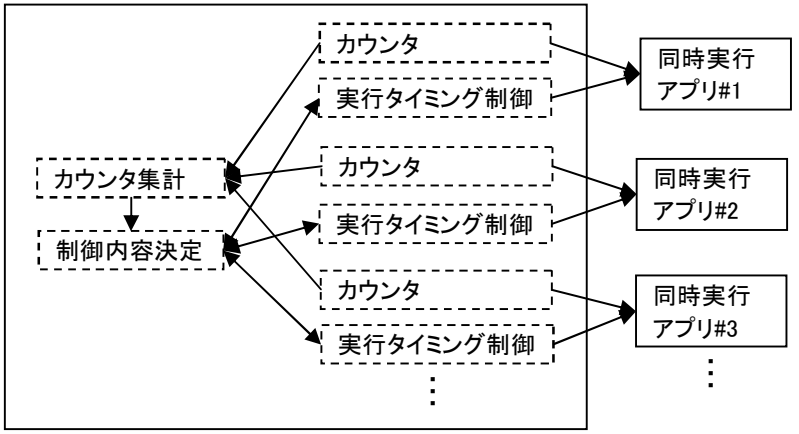


図 2 機能構成図

4. 評価システムの実装

提案するコミュニケーション品質劣化防止方式の実現可能性を確認するために、3.3 で述べた機能構成を Windows XP 上に実装した。

図 3 に示すように、カウンタ機能および実行タイミング制御機能は制御 DLL として

Vol.2010-DPS-142 No.24
Vol.2010-CSEC-48 No.24
2010/3/4

実装し、カウンタ集計機能と制御内容決定機能を独立したコミュニケーション品質制御ミドルウェアとして実装した。各制御 DLL とミドルウェア間の通信には共有メモリを利用した。制御 DLL を同時実行アプリのプロセスごとにインジェクションすることで、同時実行アプリのシステムコール利用をフックし、対応する制御 DLL の API が代わりに呼び出される。これにより制御 DLL では、システムコール実行タイミング制御を行なわない場合には、DLL では利用数をカウントするだけで、直ちに対応するシステムコールを代わりに利用することで、システムコールを直接利用したのと同じ結果を実現できる。またシステムコール実行タイミング制御中は、制御 DLL は API 呼出があってもすぐにシステムコールを利用せずにキューイング状態とし、コミュニケーション品質制御ミドルウェアからの実行指示により、実際のシステムコールを利用する。

DLL のインジェクションは既に実行中のプロセスに対しても可能なため、ソフトフォン利用のタイミングで制御 DLL をインジェクションし、制御を開始することが可能である。一方で実行ユーザが SYSTEM のプロセスの一部（alg.exe, svchost.exe）では DLL のインジェクションによる本制御ができないことも試作により明らかになった。しかし制御ができないプロセスはソフトフォン利用時に負荷が高くなることはないため、コミュニケーション品質制御としては、本実装形態は十分で有効であると言える。

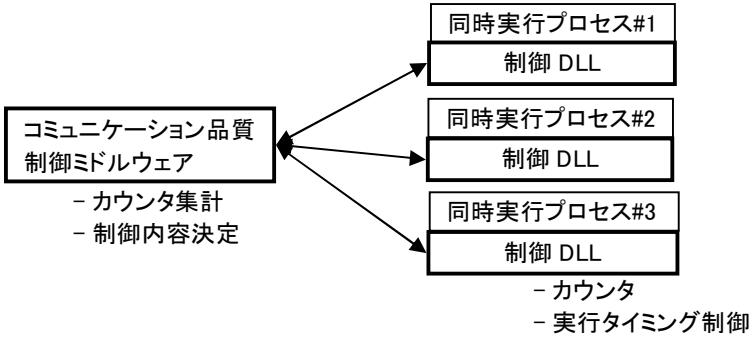


図 3 実装形態

5. 評価

本章では提案した方式が 2 章で述べた要件を満たすことを、4 章で述べた実装形態に従って試作したミドルウェアおよび DLL を用いた評価実験で確認した。

5.1 評価実験方法

5.1.1 実験システム

評価実験で用いたシステム構成を**図 4** に示す。本システムでは 2 台の PC 間でのソフトフォンでの通話中に、受信 PC で同時実行アプリケーションによる負荷を発生させ、通話音質の劣化および提案方式による音質劣化防止を確認した。

OS が Windows XP の送信 PC および受信 PC にソフトフォンをインストールし、100BASE-T の独立したネットワークで接続した。通話の代わりに送信 PC 上では音声ファイル再生してソフトフォンに対する入力とし、ソフトフォンを通じて受信 PC 側で音声を出力した。音声ファイルの長さは 1 分間で、これを実験での通話 1 回とした。通話状態の受信 PC でアンチウイルスソフトウェアおよび提案する制御ミドルウェアを実行し、アンチウイルスソフトウェアの実行による音声品質の劣化と、制御ミドルウェアによる劣化防止を確認可能とした。アンチウイルスソフトウェアは、セキュリティ系の常駐ソフトウェアでは必要不可欠なアプリケーションであり、ソフトフォン利用時に音質劣化の原因となることも多い。ユーザが通話中に意図して利用する Web ブラウザやメールクライアントと異なり、セキュリティ系の常駐ソフトウェアはユーザの意図とは関係なく動作するため、その動作に起因する音質劣化は、ユーザとしては理由も分からず突然直面するように感じられる。そのため QoE の観点から、提案方式によりアンチウイルスソフトウェアによる音質劣化が防止できることは重要であると考えられる。受信 PC の性能は**図 4** に示したとおりで、これはソフトフォンが要求する最低限のスペックである。受信 PC での音声出力はオーディオケーブルで接続した録音 PC でファイルとして録音することで、受信 PC に余計な負荷をかけることなく通話音声の劣化度合を確認できるようにした。

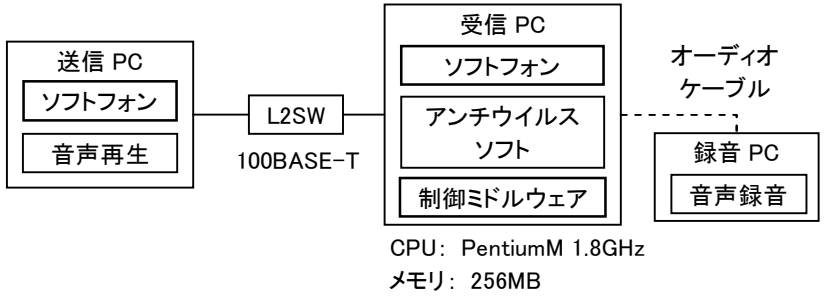


図 4 実験システム

Vol.2010-DPS-142 No.24
Vol.2010-CSEC-48 No.24
2010/3/4

5.1.2 制御対象システムコールの決定

制御対象のシステムコールを特定するため、アンチウイルスソフトウェアの動作フェーズの中で、最も音切れがよく発生するメモリスキャンについて、通話中にメモリスキャンを実行してシステムコール数を 1 秒ごとに計測し、音切れ発生時（人間による聞き取りで判定し、手動で発生時刻を記録）とそれ以外の実行時間（平常時）でのシステムコール数の変動を調査する予備実験を行った。

予備実験の結果を**図 5** に示す。音切れ発生の前後（2 秒間）と平常時で共通して実行されていたシステムコール 22 種類のうち、平常時の秒間平均実行数に対する音切れ発生時の秒間平均実行数の増加率上位 5 つをグラフで示した（平常時の秒間平均実行数を 1 としたときの割合を示す）。**図 5** から分かるように、音切れ時のシステムコールの増加率は VirtualQueryEx が突出している。VirtualQueryEx は他プロセスのメモリ情報を取得するためのシステムコールであり、メモリスキャンのためには必ず利用されるが、対象とするプロセスやメモリ状態により連続して実行されることがあり、その際に音切れが発生しているという仮説を立てることができる。そこで制御対象システムコールとして VirtualQueryEx を選択した。

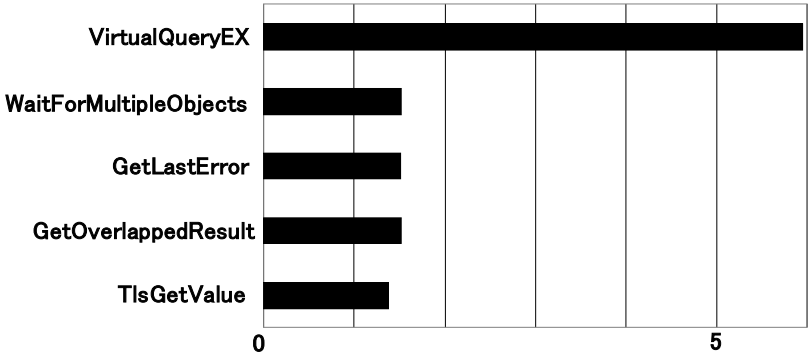


図 5 メモリスキャン中の音切れ発生時のシステムコール数増加率

5.1.3 音質評価指標

通話音声の劣化度合を客観的かつ定量的に評価するため、PESQ 値**[5]**を用いた。PESQ 値は IP 電話の音質劣化を評価するための指標値で、音声通話の送信前の音声データと受信した音声データをツールで比較することにより算出される。値は-0.5 から 4.5 の範囲となり、大きいほど音質劣化が小さい（元の音声に近い）ことを示す。**表 2** は電話サービスごとの音声品質クラスを PESQ 値で表したものである**[6][7]**。

表 2 通話サービスごとの目安となる PESQ 値

PESQ 値	3.4 以上	3.1 以上	2.5 以上
電話サービス	固定電話	携帯電話	IP 電話

5.2 評価結果

5.2.1 ソフトフォンのデータ処理の阻害の防止

要件 1 として挙げた「ソフトフォンが必要とするタイミングでコミュニケーションに関するデータ処理を実行可能にすること」が実現されていることを、ソフトフォンでの通話中にアンチウイルスソフトが動作した場合のシステムコール実行状況と音質により確認する。

実験としては、通話中にアンチウイルスソフトのメモリスキャンを実行し、提案方式による制御を行なう場合と行わない場合で VirtualQueryEx の実行状況と音質を測定した。実験は制御あり、なしでそれぞれ 20 回行った。システムコール実行間隔の初期値を 100 ミリ秒、実行待ち上限時間は 10 秒とした。

システムコール利用頻度の変化

提案方式による制御がシステムコール実行頻度を減少させていることを確認する。**図 6** にメモリスキャン中の VirtualQueryEx の毎秒実行数の最大値を示した。提案する制御により VirtualQueryEx の実行頻度は最大でも制御がない場合に比べて半減した。

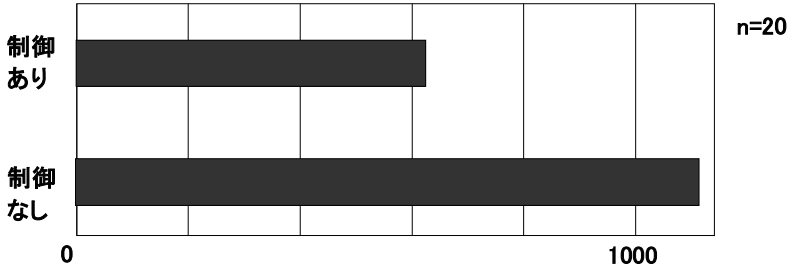


図 6 メモリスキャン中の VirtualQueryEx の最大毎秒実行数

音質劣化度合の変化

次にシステムコール実行頻度の変化が音質劣化防止に寄与していることを示す。**図 7** は提案方式による制御がある場合とない場合の、通話中に発生した音切れにおける音質（PESQ 値）の平均値と最低値である。制御の結果、固定電話の音質以下であ

Vol.2010-DPS-142 No.24
Vol.2010-CSEC-48 No.24
2010/3/4

った音切れ発生時の平均音質を元の音声を完全に再現できた場合の PESQ 値 4.5 に近づけることができた。また音質が最も劣化した場合でも、制御により固定電話の音質を実現することができた。音質の改善は、提案方式による制御によりソフトフォンが必要とするタイミングでデータ処理を実行可能になったことを意味しており、提案方式が要件 1 を満たしていることを示している。

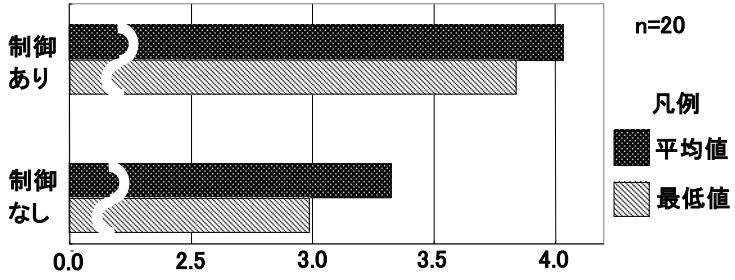


図 7 音切れ時の音質（PESQ 値）

5.3 同時実行アプリケーションの完全停止の回避

要件 2 として挙げた「ソフトフォンと同時に実行するアプリケーションを完全に停止させないこと」について、同じ実験におけるシステムコール利用数により確認する。

図 8 はメモリスキャン中にアンチウイルスソフトが実行した VirtualQueryEx の総数を、制御をした場合としない場合で比較したものである。差は総数の 0.03%しかなく、制御がメモリスキャンの実行内容には影響を与えていないことを示している。

また制御をした場合のメモリスキャン中のアンチウイルスソフトによる全システムコールの利用状況を**図 9** に示す。メモリスキャン実行中は制御されていても、アンチウイルスソフトウェアでは何らかの処理が常に実行中であることが分かる。

これらの結果より、提案方式による制御はメモリスキャンの実行内容には影響を与えず、処理を停止させないことを示しており、提案方式が要件 2 を満たしていると言える。

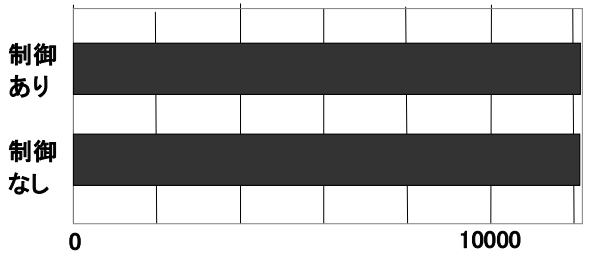


図 8 メモリスキャン中のシステムコール総数

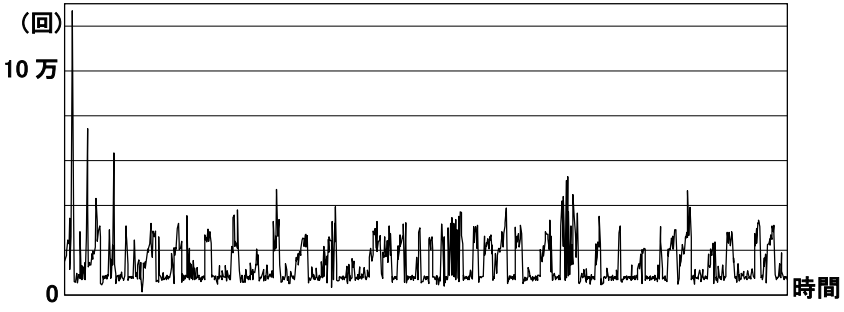


図 9 メモリスキャン中の全システムコール数の変動

6. おわりに

本稿では、様々なアプリケーションを同時に利用する PC でソフトフォンを利用したコミュニケーションを行なう場合のコミュニケーション品質劣化防止方式を提案した。本提案方式では同時実行アプリケーションのシステムコールの実行タイミングを制御することで、高優先度の OS 機能の実行が連続することを防止する。これにより同時実行アプリケーションを停止させることなく、ソフトフォンが必要なタイミングでデータ処理を実行することが可能になるため、コミュニケーション品質の劣化の発生を抑制できる。今後は提案方式を実環境に適用しての評価を行う予定である。

Vol.2010-DPS-142 No.24
Vol.2010-CSEC-48 No.24
2010/3/4

参考文献

[1] David Solomon and Mark Russinovich, “Microsoft Windows Internals, Fourth Edition: Microsoft Windows Server 2003, Windows XP and Windows 2000”, Microsoft Press

[2] Eric Eide, Tim Stack, John Regehr, and Jay Lepreau, “Dynamic CPU Management for Real-Time, Middleware-based systems” in Proceedings of the Tenth IEEE Real-time and Embedded technology and Applications Symposium (RTAS 2004)

[3] S. Brandt, G. Nutt, T. Berk and J. Mankovich, “A dynamic quality of service middleware agent for mediating application resource usage”, in Proceedings of the 19th IEEE Real-Time systems Symposium (RTSS ’98)

[4] Net Applications, “Operating System Market Share”, http://marketshare.hitslink.com/operating-system-market-share.aspx?qprid=8

[5] “Perceptual Evaluation of Speech Quality (PESQ)”, ITU-T Recommendation P.862

[6] 総務省, “「IP ネットワーク技術に関する研究会」報告書”, http://warp.ndl.go.jp/info:ndljp/pid/235321/www.soumu.go.jp/s-news/2002/020222_3.html#02

[7] “IP 電話の通話品質評価法”, TTC 標準 JJ-201.01