

監視制御システム向けアプリケーション ウィンドウクロップ伝送方式

Cropped Images of Application Window Transmission Method for Monitoring and Control System

阿倍 博信†
Hironobu Abe

川浦 健夫†
Takeo Kawaura

中島 宏一†
Koichi Nakashima

1. はじめに

監視制御システムは、コンピュータによるシステム監視とプロセス制御を行うシステムであり、対象プロセスとして、工場などの生産監視プロセス、上下水道・電力系統などのインフラ監視プロセス、ビルなどの空調やエネルギー消費などの設備監視プロセスなど広範囲にわたる。監視制御システムの最近の動向として、イーサネットの採用による通信プロトコルを含めたシステムのオープン化がある[1]。システムのオープン化により、複数ベンダの製品を組み合わせることが可能となり、単一ベンダに制限されている場合よりも拡張性、柔軟性の高いソリューションを提供できる可能性が高くなる。

しかし、現実的には全てのサブシステムを相互接続することは難しく、このような場合は、オペレータが操作するHMI (Human Machine Interface) コンピュータを複数台設置する必要が出てきてしまう、といった問題があった。

このような背景の元、本稿では、オープンプラットフォームのリモートデスクトップ技術の一つであるVNC (Virtual Network Computing) [2]を用いた監視制御システム向けアプリケーションウィンドウのクロップ伝送方式を提案する。

VNCをベースとすることで、プラットフォームに依存しない形でリモートデスクトップ環境を実現でき、本方式により、他のサブシステムのHMIコンピュータ上で動作している任意のアプリケーションのウィンドウ画面をウィンドウ単位で切り出して伝送し、ローカルのHMIコンピュータのデスクトップ上に統合表示して遠隔操作することが可能となる。

また、提案方式の利点としては、リモートコンピュータのデスクトップ画面を必要なアプリケーションのウィンドウ単位で切り出して伝送することで、デスクトップ画面全体を伝送する場合と比較して、操作性の向上及び通信トラフィックの効率化が考えられる。

本稿では、2章で関連研究、3章でアプリケーションウィンドウのクロップ伝送方式の提案、4章で評価システムの開発、5章で評価について述べ、最後にまとめを行う。

2. 関連研究

従来の監視制御システムのHMIコンピュータでは、OSとしてはUnixが、デスクトップ環境としてはX Window System[3]が中心として採用されていた。Unix/X Window Systemの単一環境では、コンピュータ間でのウィンドウ単位での画面データの相互伝送が可能であった。しかし、HMIコンピュータのOSとしてWindowsが採用されるよう

になり、現在では、Unix/X Window Systemプラットフォームとの混在が前提となる。

本稿では、プラットフォームに依存しないリモートデスクトップ技術であるVNCに着目する。VNCベースのローカル/リモートのデスクトップ統合を目的とした関連研究としてMetaVNC[4]があげられる。MetaVNCは、アプリケーションのウィンドウ以外の領域を透過領域として扱うことにより、複数のリモートデスクトップ画面を単一デスクトップに合成して表示する機能を持つ。

MetaVNCの現状の実装では、リモートデスクトップ単位で1枚のレイヤを構成する構造となっているため、監視制御システムのHMIコンピュータに適用する場合に以下の技術課題が考えられる。

(1) 操作性の改善

リモートコンピュータ上で複数のアプリケーションウィンドウが動作している際に、あるリモートウィンドウをクリックすると、注目していない他のウィンドウと一緒に前面に表示されてしまう。

(2) 通信トラフィックの改善

VNCは基本的にデスクトップ全体を伝送する方式のため、リモートコンピュータ上でリモート伝送の必要の無いアプリケーションで画面更新頻度の高いものが動作している場合、不要な通信トラフィックが発生してしまう。

3. アプリケーションウィンドウのクロップ伝送方式の提案

3.1 設計指針

2章で設定した課題を解決することを目的として、リモートコンピュータのデスクトップ画面からユーザの指定したアプリケーションのウィンドウ画面を切り出して伝送し、ローカルコンピュータ上のデスクトップに統合表示する方式について提案する。

提案方式の設計指針について下記の通り設定する。

(1) 既存方式の拡張として実現できること

既存方式の拡張方式として必要機能のみ実装することで、VNCのソフトウェア資産の有効活用が図れる。

(2) プラットフォーム依存しない方式であること

リモート/ローカルのOSがWindows/Unixのどちらであっても対応可能である。

3.2 提案

3.1節で設定した設計指針に従い、リモートコンピュータのデスクトップ画面の指定アプリケーションのウィンドウイメージをクロップ伝送する方式について提案する。提案方式は3ステップから構成され、下記にステップごとに内容を説明する。

(1) ステップ1: ウィンドウ一覧取得

†三菱電機株式会社, Mitsubishi Electric Corporation

リモートコンピュータ上で、メインウィンドウを持ったアプリケーションのウィンドウ一覧を取得する。取得方式は、Windows と Unix/X Window System では異なるが、OS のシステムコール経由で取得する。

(2) ステップ2: 対象ウィンドウの指定

ステップ1で取得したウィンドウ一覧からクロップ伝送したいウィンドウを指定する。指定方式はGUIによる指定、ネットワーク経由での指定、設定ファイルによる指定のどれかとする。

(3) ステップ3: 画面イメージの伝送

ステップ2で指定したウィンドウの切り出した画面イメージをVNCサーバの伝送対象として指定する。ローカルコンピュータでVNCビューワを起動し、ネットワーク経由でVNCサーバに接続すると、指定されたウィンドウの画面イメージのみ伝送・表示される。

4. 評価システムの開発

3章の提案内容を評価するための評価システムを開発した。まず、ベースシステムとして、マルチプラットフォームに対応したTightVNC[5]を選定した。

リモートコンピュータ上で、TightVNCサーバとの連携によりステップ1~3を実行する機能を持った画面クロップ伝送アプリを開発した。

リモートコンピュータとネットワーク接続されたローカルコンピュータで起動したTightVNCビューワからTightVNCサーバに接続すると、指定されたウィンドウの画面イメージのみが伝送・表示される。

開発した評価システムの仕様について表1に示す。

表1 評価システムの仕様

| | リモートコンピュータ | ローカルコンピュータ |
|-----|---|-------------------------|
| S/W | TightVNC Server (1.3.10) 画面クロップ伝送アプリ | TightVNC Viewer (1.5.2) |
| OS | Windows XP SP3 | Windows XP SP3 |
| 解像度 | 1024×768 (32bpp) | 1280×800 (16bpp) |
| CPU | Pentium M 1.40GHz | Core2Duo 2.4GHz |

5. 評価

4章で開発した評価システムを用いてシステムの有効性について下記の評価を実施した。

操作性については、提案方式であるリモートコンピュータにて指定したアプリケーションのウィンドウの画面イメージのみを切り出してローカルコンピュータに伝送・表示できていることが確認でき、MetaVNCと比較して操作性の向上が確認できた。

通信トラフィックの評価にあたり、リモートコンピュータ上で対象アプリケーションとしてトラフィックモニタ(画面更新頻度50msec)とパフォーマンスモニタ(画面更新頻度1sec)を動作させ、その際のローカルコンピュータの平均送信ビットレート、平均受信ビットレート、リモートコンピュータの平均CPU使用率を測定した。この評価において、伝送対象は「(1)デスクトップ全体、(2)トラフィックモニタのみ、(3)パフォーマンスモニタのみ」から選択し、画面データの圧縮形式は「Raw, Hextile, Tight, RRE, CoRRE, Zlib(pure), ZlibHex(mix)」から選択するようにした。

性能評価の結果を表2に示す。

表2の結果について、下記の通り考察する。

(1) 全画面伝送の場合、圧縮形式をTight, Zlibを選択すると他の圧縮形式より送信/受信ビットレートを低くできるが、その分CPU負荷が高くなる。

(2) トラフィックモニタとパフォーマンスモニタを動作させ、トラフィックモニタのみをクロップ伝送した場合は、画面全体を伝送した場合と比較して若干送信/受信ビットレートやCPU負荷が低下するだけであるが、パフォーマンスモニタのみをクロップ伝送した場合は、画面全体を伝送した場合と比較して送信/受信ビットレートやCPU負荷が大幅に低下するとともに、通信トラフィックの改善が確認できた。

6. おわりに

本稿では、監視制御システムにおけるプラットフォームの異なる複数サブシステム間でのHMIコンピュータ間の統合を目的として、VNCを活用してリモートコンピュータ上で指定したアプリケーションのウィンドウイメージを切り出してローカルコンピュータに伝送・表示する方式について提案を行った。評価システムを開発して評価を行った結果、システムの有効性について確認することができた。

参考文献

- [1]情報処理推進機構：重要インフラの制御システムセキュリティとITサービス継続に関する調査(2009)。
- [2]VNC - Virtual Network Computing from AT&T Laboratories Cambridge: <http://www.cl.cam.ac.uk/research/dtg/attarchive/vnc/>
- [3]X.Org Foundation: <http://www.x.org/wiki/>
- [4]MetaVNC - a window aware VNC: <http://metavnc.sourceforge.net/index.html>
- [5]TightVNC: VNC-Compatible Free Remote Control/Remote Desktop Software : <http://www.tightvnc.com/>

表2 性能評価結果

| Encoding Type | FullScreen (1024x768x32bpp) | | | CroppedScreen1 (778x433x32bpp) | | | CroppedScreen2 (641x453x32bpp) | | |
|---------------|-----------------------------|-----------|---------|--------------------------------|-----------|---------|--------------------------------|-----------|---------|
| | 送信 (kbps) | 受信 (kbps) | CPU (%) | 送信 (kbps) | 受信 (kbps) | CPU (%) | 送信 (kbps) | 受信 (kbps) | CPU (%) |
| Raw | 174 | 12031 | 26 | 165 | 11491 | 23 | 4 | 47.1 | 18 |
| Hextile | 162 | 7942 | 38 | 159 | 7704 | 34 | 3.4 | 6.7 | 18 |
| Tight | 40 | 1152 | 58 | 39 | 1155 | 54 | 3.6 | 8.6 | 19 |
| RRE | 109 | 7192 | 29 | 123 | 8262 | 27 | 3.4 | 8.6 | 18 |
| CoRRE | 127 | 7810 | 32 | 140 | 8179 | 28 | 3.5 | 9.7 | 19 |
| Zlib(pure) | 40 | 943 | 46 | 39 | 931 | 44 | 3.6 | 6.5 | 19 |
| ZlibHex(mix) | 70 | 2888 | 52 | 73 | 2970 | 52 | 3.5 | 9 | 19 |

Cropped Screen1: トラフィックモニタ, Cropped Screen2: パフォーマンスモニタ